

# Performance Analysis of Constant Speed Local Abstacle Avoidance Controller Using a MPC Algorithm on Granular Terrain

Nicholas Haraus  
*Marquette University*

---

## Recommended Citation

Haraus, Nicholas, "Performance Analysis of Constant Speed Local Abstacle Avoidance Controller Using a MPC Algorithm on Granular Terrain" (2017). *Master's Theses (2009 -)*. 443.  
[http://epublications.marquette.edu/theses\\_open/443](http://epublications.marquette.edu/theses_open/443)

PERFORMANCE ANALYSIS OF A CONSTANT SPEED LOCAL OBSTACLE  
AVOIDANCE CONTROLLER USING A MPC ALGORITHM  
ON GRANULAR TERRAIN

by

Nicholas Haraus, B.S.M.E.

A Thesis submitted to the Faculty of the Graduate School,  
Marquette University,  
in Partial Fulfillment of the Requirements for  
the Degree of Master of Science

Milwaukee, Wisconsin

December 2017

ABSTRACT  
PERFORMANCE ANALYSIS OF A CONSTANT SPEED LOCAL OBSTACLE  
AVOIDANCE CONTROLLER USING A MPC ALGORITHM  
ON GRANULAR TERRAIN

Nicholas Haraus, B.S.M.E.

Marquette University, 2017

A Model Predictive Control (MPC) LIDAR-based constant speed local obstacle avoidance algorithm has been implemented on rigid terrain and granular terrain in **Chrono** to examine the robustness of this control method. Provided LIDAR data as well as a target location, a vehicle can route itself around obstacles as it encounters them and arrive at an end goal via an optimal route. This research is one important step towards eventual implementation of autonomous vehicles capable of navigating on all terrains. Using **Chrono**, a multibody physics API, this controller has been tested on a complex multibody physics HMMWV model representing the plant in this study. A penalty-based DEM approach is used to model contacts on both rigid ground and granular terrain. Conclusions are drawn regarding the MPC algorithm performance based on its ability to navigate the **Chrono** HMMWV on rigid and granular terrain. A novel simulation framework has been developed to efficiently simulate granular terrain for this application.

Two experiments were conducted to analyze the performance of the MPC LIDAR-based constant speed local obstacle avoidance controller. In the first, two separate controllers were developed, one using a 2-DOF analytical model to predict the HMMWV behavior, and the second using a higher fidelity 14-DOF vehicle model. In this first experiment, two controllers were compared as they controlled the HMMWV on two obstacle fields on rigid ground and granular terrain to understand the influence of model fidelity and terrain on controller performance. From these results, an improved lateral force model was developed for use in the 2-DOF vehicle model to better model the tire ground interaction using terramechanics relations. A second experiment was performed to compare two developed controllers. One used the 2-DOF vehicle model using the Pacejka Magic Formula to estimate tire forces while the second used a 2-DOF vehicle model with the newly developed force model to estimate lateral tire forces.

As a result of this research, a smarter controller was developed that uses friction angle, cohesion, and interparticle friction coefficient to more accurately predict vehicle trajectories on granular terrain and allow a vehicle to navigate autonomously on granular terrain.

## ACKNOWLEDGEMENTS

Nicholas Haraus, B.S.M.E.

Obtaining a degree has not been a solo pursuit, and therefore I would like to express my deep appreciation and gratitude in no particular order to:

My loving family, for their continued support of me throughout my life and educational journey.

Sara, for her love, care, support, humor, and encouragement to continue pushing forward even in the most stressful times.

My close friends, for the distractions and adventures that made my time at Marquette all the more memorable.

My adviser, Dr. Jonathan Fleischmann, for the opportunity to work with him for the past few years, enabling me to grow as a student, researcher, engineer, and person.

Dr. Philip Voglewede, for the technical expertise, advice, and for teaching me...and reteaching me...and reteaching me....that spring mass damper systems are in fact everywhere.

All the faculty and staff at Marquette for providing an incredible educational experience.

Dr. Radu Serban and Dr. Dan Negrut from the Simulation Based Engineering Laboratory in Madison, for their support and collaboration throughout this research endeavor.

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b>	<b>i</b>
<b>LIST OF TABLES</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>v</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Summary . . . . .	2
1.2 Literature Review . . . . .	4
1.2.1 Early Obstacle Avoidance Algorithms: Artificial Potential Field, Vector Field Histogram, and Dynamic Window Approaches . .	4
1.2.2 MPC Algorithms . . . . .	11
1.2.3 MPC Algorithms for Car-Like Vehicle . . . . .	14
1.2.4 Extending Rigid Ground Tests to Granular Terrain . . . . .	17
1.2.5 Simulating Granular Terrain for Vehicular Applications . . . .	18
1.2.6 Summary . . . . .	20
1.3 Objectives and Methodology . . . . .	20
<b>2 THEORETICAL BACKGROUND</b>	<b>23</b>
2.1 Multibody Dynamics Background . . . . .	23
2.2 Dealing with Contacts: The Penalty Method . . . . .	25
2.3 Chrono Multibody Physics Package . . . . .	28
2.3.1 Vehicle Modeling . . . . .	28
2.4 MPC LIDAR-Based Local Obstacle Avoidance: History and Current Algorithm . . . . .	29
2.5 Multi-DOF Vehicle Dynamics Models . . . . .	34
2.5.1 2-DOF Vehicle Model . . . . .	35
2.5.2 14-DOF Vehicle Model . . . . .	36

2.6	Tire Ground Force Model: Pacejka Magic Formula . . . . .	37
2.7	Optimization Problem: Exhaustive Search Space . . . . .	39
<b>3</b>	<b>EXPERIMENTAL SETUP</b>	<b>43</b>
3.1	Controlled Chrono HMMWV Vehicle . . . . .	43
3.2	Simulation Parameters . . . . .	45
3.3	Evaluation Metrics . . . . .	47
3.4	Granular Terrain Simulation . . . . .	47
3.5	Hardware and Materials . . . . .	54
<b>4</b>	<b>FIRST EXPERIMENTAL RESULTS AND DISCUSSION</b>	<b>56</b>
4.1	Influence of Model Fidelity . . . . .	57
4.2	Influence of Terrain Type . . . . .	58
<b>5</b>	<b>IMPROVED VEHICLE MODEL ON GRANULAR MATERIAL</b>	<b>61</b>
5.1	Bekker/Wong Load Sinkage Relationship . . . . .	62
5.2	New Simplified Force Model . . . . .	63
5.3	Open Loop Comparisons . . . . .	66
<b>6</b>	<b>SECOND EXPERIMENTAL RESULTS TO TEST IMPROVED VEHICLE MODEL</b>	<b>70</b>
6.1	Experimental Setup . . . . .	70
6.2	Results . . . . .	71
<b>7</b>	<b>CONCLUSIONS</b>	<b>75</b>
7.1	Future Work . . . . .	80
	<b>BIBLIOGRAPHY</b>	<b>84</b>
	<b>APPENDIX A: RAY-CIRCLE INTERSECTION ALGORITHM</b>	<b>88</b>
	<b>APPENDIX B: POINT IN POLYGON ALGORITHM</b>	<b>91</b>

LIST OF TABLES

3.1	Individual Simulation Test Information . . . . .	45
3.2	Obstacle Field Parameters . . . . .	46
3.3	Hardware Components of Computer Used for Parallel Simulations . . . . .	55
4.1	Experiment 1 Test Evaluation Metrics Summary on Obstacle Field 1	56
4.2	Experiment 1 Test Evaluation Metrics Summary on Obstacle Field 2	57
6.1	Compared Simulations for Second Experiment . . . . .	70
6.2	Experiment 2 Test Evaluation Metrics Summary on Obstacle Field 1	71
6.3	Experiment 2 Test Evaluation Metrics Summary on Obstacle Field 2	72

## LIST OF FIGURES

1.1	Comparison from Literature Review of Artificial Potential Field, Vector Field Histogram, and Dynamic Window Approaches to local obstacle avoidance . . . . .	10
2.1	Reproduced from [1]. DEM-P contact model with normal overlap distance $\delta_n$ , contact-normal unit vector $\mathbf{n}$ , and tangential displacement vector $\mathbf{u}_t$ (top), and a Hookean-linear contact force-displacement model with constant Coulomb sliding friction (bottom) . . . . .	26
2.2	Schematic of MPC LIDAR-Based Constant Speed Local Obstacle Avoidance Controller . . . . .	30
2.3	Sample obstacle field and LIDAR output . . . . .	31
2.4	2-DOF Vehicle Model . . . . .	36
2.5	Reproduced from [2]. 14-DOF Vehicle Model . . . . .	37
2.6	Exhaustive Search Space Simplified Visualization . . . . .	40
2.7	Exhaustive Search Space Paths Checked with Point in Polygon Algorithm . . . . .	41
3.1	Full-vehicle HMMWV multibody model. . . . .	44
3.2	Obstacle Fields . . . . .	46
3.3	Relocating Granular Patch that follows the vehicle . . . . .	48
3.4	Granular relocation algorithm . . . . .	49
3.5	HMMWV navigating around obstacle, granular Patch does not collide with obstacles . . . . .	51
3.6	Independent four granular relocating patches explored to improve simulation speeds, HMMWV chassis not visualized . . . . .	53
4.1	Experiment 1 Test results on Obstacle Field 1 . . . . .	56
4.2	Experiment 1 Test results on Obstacle Field 2 . . . . .	57
5.1	Terrain interactions on a single wheel on soft soil . . . . .	62
5.2	Comparison of Magic Formula Lateral Force Prediction to New Force Model . . . . .	64
5.3	Effect of varying internal friction angle $\phi$ on lateral force prediction . . . . .	65

5.4 Steering Sequence used for Open Loop Simulations . . . . . 67

5.5 Open Loop Trajectory of HMMWV over non-rotating granular terrains with  
different inter-particle friction coefficients  $\mu$  . . . . . 68

5.6 Comparison of Trajectory Predicted by 2-DOF model using the Magic For-  
mula to estimate Tire Forces and the 2-DOF model using the new simplified  
force model . . . . . 68

6.1 Experiment 2 Test results on Obstacle Field 1 . . . . . 71

6.2 Experiment 2 Test results on Obstacle Field 2 . . . . . 72

7.1 HMMWV unable to make forward progress in 6-DOF particle field . . . . . 82

9.2 Ray-Circle Intersection Visualization . . . . . 88

9.3 Visualization of Point in Polygon Algorithm Working Successfully . . . . . 91

## 1 INTRODUCTION

Obstacle avoidance is a crucial capability for Autonomous Ground Vehicles (AGVs) of the future. This refers to a ground vehicle's ability to sense its surrounding environment, develop an optimal path around the obstacles in the environment, generate optimal control commands to satisfy that path, and physically navigate around the obstacles safely and to a desired endpoint. Safety is defined as avoiding collisions as well as enforcing limitations on excessive sideslip or tire lift-off. An ideal control algorithm is one that is capable of pushing a vehicle to its performance limits by using knowledge of its dynamic capabilities and surrounding environmental conditions, while still enforcing strict safety requirements. Although previous work has demonstrated use of Model Predictive Control (MPC) algorithms for obstacle avoidance on wheeled vehicles, more work is required to test the fidelity of these algorithms and determine where improvements are needed.

One area in which MPC algorithms have yet to be tested is their ability to control a wheeled vehicle on granular terrain. Up to this point, the terrain has been assumed to be rigid and flat. The performance of MPC algorithms on deformable terrain raises an additional question. Do the current most commonly used vehicle models perform successfully within an MPC algorithm on deformable terrain? The present work evaluates, through numerical simulation, the robustness and validity of an MPC algorithm with different vehicle models in an environment more similar to what an off-road military vehicle would experience in combat. This controller could be applied to any car-like wheeled vehicle.

The remainder of this thesis is organized as follows. The remainder of Chapter 1 consists of the problem statement, a thorough literature review into the problem, and the objectives of this research based on the findings of the literature review.

Chapter 2 consists of theoretical background and overviews pertinent to this research. Chapter 3 describes the Chrono::Vehicle HMMWV model used in this study as the plant model to be controlled and the proposed method for simulating a vehicle driving on deformable terrain over large distances. Chapter 4 presents the results of the tests performed in Experiment 1 and provides comparisons when the internal controller vehicle model is varied and when the terrain is changed from rigid to granular. Chapter 5 presents a new intermediate vehicle model that uses terramechanics relations to better predict vehicle trajectories on granular terrain. Chapter 6 contains the description and results of Experiment 2 performed to test the new vehicle model within the MPC controller. Chapter 7 wraps up the thesis and presents potential future work based on the results of this study.

## 1.1 Problem Summary

Regarding the MPC control algorithm, many studies have been performed previously to validate its potential as a local obstacle avoidance controller. Specifically, studies by Ersal, et al. explored the implementation of an MPC local obstacle avoidance algorithm in AGVs. The research group tested a steering controller for an AGV on rigid flat terrain [3], then a combined steering and speed controller [4], and finally studied the role of model fidelity in the implementation of the MPC algorithm [5] to understand how detailed the internal controller model needs to be to effectively and quickly control the vehicle. However, in these studies and those performed before them, the AGV has been tested on rigid flat terrain, an assumption which does not hold when driving a vehicle in reality. Therefore, the research for this thesis will perform studies on the promising models from [5] and test them on granular terrain to understand how changing the ground from rigid and flat to granular affects the controlled AGV performance.

This research also addresses another larger problem for the military. The North Atlantic Treaty Organization (NATO) is seeking to replace their current NATO

Reference Mobility Model (NRMM) [6]. The current NRMM is an empirically based computer model developed during the 1970s, which computes tactical mobility metrics by correlating engineering level performance of various ground vehicle systems with different terrain conditions, allowing for successful comparison of these ground systems and capabilities over varied terrain surfaces, obstacles, vegetation, weather scenarios, grades, and other features which can adversely impact ground vehicle performance. NATO seeks to replace this older model with a new physics-based computer simulation solution which can still answer the same questions as the older model, but can then be extended past situations with only empirical results. One of the key areas of work in this group effort is the ability to develop and deliver to the vehicle go-nogo maps of upcoming terrain, either from recent satellite imagery or other sorts of sensors. With this research, the development of this simulation and test will seek to control an AGV only with the sensor data or terrain inputs provided by the mission planner. Any research achieved relating to vehicle mobility across a variety of terrains can help with the NRMM effort.

Significant studies have been carried out and work accomplished testing the MPC Algorithm and validating **Chrono** as a viable program for Multibody simulations. However, the two topics have not combined to be tested together. The MPC algorithm has been tested on rigid flat terrain, yet with **Chrono::Parallel** one has the ability to simulate granular terrain if desired with reasonable runtimes. Therefore, one can develop a simulation of a vehicle on granular terrain with scattered obstacles in **Chrono::Parallel**, and then use the MPC algorithm from [5] to control that simulated vehicle. Doing so would benefit both the simulations and controls fields. By bridging the gap between these two areas, those designing the control algorithms have a quicker and less expensive method of testing a control algorithm on a full vehicle on a variety of landscapes to gain quick understanding of

how well their controller behaves. On the other hand, developing this sort of simulation improves the amount of applications `Chrono::Parallel` can be used in. This research has performed higher fidelity testing of the control algorithm than has previously been accomplished, a step which is necessary if this algorithm is to eventually be implemented in an actual vehicle.

The uniqueness of this research comes from two areas: controls and simulation. Most `Chrono` simulations created so far do not test a complex control algorithm for an autonomous vehicle. More interest is placed in understanding vehicle mobility over different terrains. However, this research tests an autonomous vehicle and its ability to navigate around local sensed obstacles. Therefore, a framework has been developed to allow the simulated vehicle to sense obstacles in the area and generate its own commands while it navigates across the terrain instead of relying on externally supplied commands. Can this MPC control algorithm generate the appropriate commands to navigate a vehicle through an obstacle field? On the controls side, this is a unique effort as well. As mentioned, the MPC algorithm has yet to be tested on granular terrain. However, considering an AGV will be driving off road on soil or sand, this is an appropriate next step for testing. This research answers the question of if simulation can be used to effectively test an autonomous vehicle on granular terrain. Computationally this requires a lot of resources, but if a team can test their control algorithm virtually before going out to the test field with a vehicle, this could save both time and money for that team.

## 1.2 Literature Review

### 1.2.1 Early Obstacle Avoidance Algorithms: Artificial Potential Field, Vector Field Histogram, and Dynamic Window Approaches

Many obstacle avoidance algorithms have been developed in the past that allow for smooth, continuous, fast motion of an AGV through an obstacle field. Early

algorithms were primarily developed and tested on small ground robots and mainly focused on finding a collision free path. However, they did not necessarily guarantee optimality or the vehicle's ability to achieve that prescribed collision-free path. Some early researched algorithms were developed using artificial potential methods. Applied to a manipulator moving through space, the philosophy of the potential field approach can be summarized as: "The manipulator moves in a field of forces. The position or target to be reached is an attractive pole while all obstacles are repulsive surfaces for the manipulator parts" [7]. Khatib described the formulation and implementation of the artificial potential field concept towards obstacle avoidance. Using the kinematic relationships of the system to be controlled, a Lagrangian formulation is used to develop the equations of motion of the system. An artificial potential field is created that is non-negative continuous and differentiable whose value goes to infinity as the vehicle approaches an obstacle. The overall artificial potential function is then a sum of the contributing potential fields caused by each of the obstacles within the obstacle field. Each of the obstacles are modeled as a composition of primitives with analytical equations representing their collision envelopes. Khatib successfully implemented this obstacle avoidance algorithm real-time in an experimental manipulator programming system *Control in Operational Space of a Manipulator with Obstacles System* (COSMOS) with links and moving obstacles [7].

Rimon and Koditschek then presented a technique for constructing artificial potential fields that could bring a bounded-torque actuated robot to a desired configuration without hitting obstacles [8]. Their formulation works for any n-DOF robot whose configuration space happens to be a generalized sphere world. However as with [7], this formulation requires *a priori* knowledge of the topology of the obstacle field. Therefore the implementation of the algorithm at the point of publication of [8] is infeasible for unknown obstacle fields.

Vector Field Histogram methods were then researched due to their ability to navigate a robot through an unknown environment where topology is not known *a priori*. The Vector Field Histogram (VFH) method allows the detection of unknown obstacles and avoids collisions as it steers a mobile robot to some end destination [9]. The VFH method models the world as a two-dimensional Cartesian histogram grid and continuously updates the grid based on the input on-board sensor data. The algorithm reduces the histogram to a one-dimensional polar histogram around the current robot location such that each sector of the polar histogram has a polar obstacle density associated with it. The algorithm then selects the sector with a low obstacle density and aligns the robot's steering with that sector [9]. The VFH method outlined in [9] successfully navigated a mobile robot through an obstacle field at an average speed of 0.6-0.7 m/s. This method is a local path planner, so there is no way of it purposefully following a globally optimal path. This algorithm and artificial potential field methods are susceptible to dead end situations and local minima [9].

In a much more recent study, Gong and Duan proposed a multi-strategy navigation method to combat some common problems with autonomous vehicle navigation [10]. The two common problems addressed were the vehicle reaching a local minimum and the vehicle navigating into a dead end. Both scenarios result in the vehicle remaining stuck forever with many algorithms. The proposed multistrategy algorithm identifies the current state of the vehicle and determines which navigation strategy is best for the current moment. Vector Polar Histogram is used until the vehicle gets stuck at a dead end or local minimum. The controller then switches modes to a wall-following algorithm, forcing the vehicle to find a wall and follow it until out of the dead end. If the vehicle is stuck in a local minimum, the controller may switch to a move-towards-goal algorithm forcing the vehicle to progress straight to the goal if not blocked by obstacles for a short time until away

from the local minimum. The move-towards-goal algorithm would also apply to the situation where a vehicle's path to the goal is clear, but the goal is right next to an obstacle and therefore the Vector Polar Histogram prevents that path from being considered as a possibility. This method successfully navigated a vehicle in both simulation and real life at a top speed of 1 m/s [10].

Dynamic Window approaches at their roots are most similar to the MPC algorithm used for the research of this thesis. Fox, Burgard, and Thrun focus specifically on the reactive avoidance of collisions with obstacles by a robot [11]. A dynamic window approach is proposed that deals with constraints imposed by velocity and acceleration limits. Steering commands are computed periodically thus avoiding the complexity of a general motion planning problem. First, they consider velocities that result in circular trajectories. Then, only velocities that can be reached within the next time window are considered, largely decreasing the search space forming the dynamic window. Admissible velocities are weighed by an objective function. In experiments, this method successfully navigated a B21 mobile robot at a top speed of 0.95 m/s [11].

Brock and Khatib then proposed a global dynamic window approach to obstacle avoidance that combines path-planning and real-time obstacle avoidance algorithms to generate robot motions to complete a task and still remain safe in an unknown environment [12]. As the robot moves through the environment, it builds an occupancy grid based off sensor input data to represent the connectivity of free space, allowing the robot to learn about its environment without any global *a priori* knowledge of the obstacles. The global dynamic window approach combines the reactive collision avoidance of the dynamic window approach of [11] with a global, local minima-free navigation function *NF1*. An objective function is developed to choose the best path forward. This global dynamic window approach was successfully tested with a synchro-driven mobile robot Nomad XR4000 at 1.0 m/s

[12].

Kunchev, Jain, Ivancevic, and Finn presented a review and comparison of path planning and obstacle avoidance techniques including those previously mentioned. Each obstacle avoidance method has its own benefits and disadvantages. The issue of local minima is significant for all developed obstacle avoidance algorithms. All the mentioned techniques need more development to be applied to a car-like vehicle at higher speeds [13]. More recent research aims to take these early collision avoidance algorithms and adapt them towards the navigation of high speed AGV's. The motion of these AGV's is not as simple as the small robots the algorithms were initially developed for in that the AGV cannot instantaneously move in any direction. Ignoring the effects of slip, a car-like AGV will move in its heading direction. Newer research has aimed to build upon or even combine previously mentioned algorithms for the application towards car-like AGV's.

Shimoda, Kuroda, and Iagnemma proposed a potential field based method of navigating an unmanned ground vehicle at high speeds across sloped and rough terrains [14]. With this proposed method, a potential field is generated in a two dimensional trajectory space of the path curvature and longitudinal velocity. The potential field is generated based on dynamic constraints, terrain slopes, obstacle proximities, and the target location. A maneuver is selected within a set of performance bounds based on the local potential field gradient. Each maneuver is mapped to the low level commands necessary to execute that maneuver. This method has successfully been tested at a speed of 7 m/s [14].

Koren and Borenstein then performed a study presenting a systematic overview of the downfalls of potential field methods [15]. Four main drawbacks are noted and described. First, a well-documented issue is trap situations due to local minima. Second, even though the controlled robot may be slightly smaller than a certain gap between two obstacles, the combined repulsive force from the two obstacles prevents

the robot from passing through gaps it should be able to. Third, the robot motion exhibits oscillatory behavior near obstacles. This same behavior is seen in the fourth noted downfall where if a robot is moving through a narrow-walled passage, the robot will oscillate close to each wall throughout the whole passage again due to the nature of the developed potential field and repulsive forces. For certain applications, potential field methods may be appropriate due to simple, elegant, and quick navigation of the robot through the obstacle field. However, Routh stability criterion presented in [15] prove this method would not be stable as vehicle mass and speed increase. Overall these drawbacks make the artificial potential field method unsuitable for the research of this study. These drawbacks have been experimentally proven and motivated a move by Koren and Borenstein away from potential field methods [15].

A summary and comparison of the explored Artificial Potential Field, Vector Field Histogram, and Dynamic Window Approach are presented in Figure 1.1. Each of the methods has their own advantages and disadvantages. The Artificial Potential Field and VFH methods are both quick and simple which is desirable for realtime implementation. Potential field methods however often rely on known *a priori* information about the environment which will not always be available. The VFH method is quick, simple, and allows the vehicle to navigate around obstacles as it encounters them, but provides limited support for inclusion of safety constraints and obstacle avoidance of more complex systems such as car-like vehicles. The Dynamics Window approaches, though more complex, support the inclusion of multiple safety related constraints, solve an optimal control problem over a smaller finite time window, and can be applied to multiple different systems so long as they can be modeled mathematically.

	<b>Artificial Potential Field</b>	<b>Vector Field Histogram</b>	<b>Dynamic Window Approach</b>
<b>Overview</b>	creates an artificial potential field that approaches infinity near obstacles and zero at the desired target point, vehicle follows minimum gradient path	models world using a 2D cartesian histogram grid and updates grid based on input on-board sensor data, vehicle moves in a direction of low obstacle density	solves optimal control problem over finite time horizon throughout navigation, generates steering commands periodically, uses safety constraints on velocity and acceleration to determine an optimal path using an objective function
<b>Benefits</b>	quick if environment is known <i>a priori</i> , ideal for small and slow robots, simple	no <i>a priori</i> information about environment necessary, sensor-oriented, simple	handles addition of constraints well, flexibility in defining "optimality" and decreasing search space, sensor-oriented, no <i>a priori</i> obstacle information necessary, easily applicable to variety of mechanical systems so long as one can model the system dynamics
<b>Negatives</b>	most algorithms require <i>a priori</i> knowledge of obstacle field and environment, issues with local minima, trouble navigating tight fit gaps, oscillatory behavior near obstacles and in narrow walled passages, not stable as vehicle mass and speed increase, difficult to generate potential function quickly during navigation, paths not optimal	non-optimal paths, developed for small slow synchro-driven robots, difficult to impose multiple different constraints to maintain vehicle safety	computationally expensive, limits on vehicle speed based on hardware capabilities of controller, non-convex stability criteria

Figure 1.1: Comparison from Literature Review of Artificial Potential Field, Vector Field Histogram, and Dynamic Window Approaches to local obstacle avoidance

### 1.2.2 MPC Algorithms

To address the issue of optimality, more rigorous approaches have been developed leveraging the Model Predictive Control approach. With this method, control action is obtained by solving a finite horizon open-loop optimal control problem over a receding horizon.

Findheisen and Allgöwer provide an introduction to nonlinear Model Predictive Control [16]. Nonlinear MPC methods are being explored due to the method's ability to incorporate design driven constraints into the robot control. As time goes on, systems are becoming subject to more safety and environmental constraints making the application of previously mentioned methods difficult. With nonlinear MPC, it is not too difficult to add more system constraints to the optimal control problem solved each time step. Multiple applications and derived nonlinear MPC methods are presented as well in [16]. MPC is therefore a promising approach for obstacle avoidance due to its capability to handle input saturation, system nonlinearities, and design state constraints in a dynamic environment.

Ogren and Leonard present the Dynamic Window Approach for fast and safe obstacle avoidance in unknown environments and recast the approach in a continuous nonlinear framework, drawing many similarities to MPC [17]. A method for generating a navigation function with a single unique minimum is presented. In this study, the control input space is discretized for a computationally tractable version of MPC, and exhaustive search is used to identify the best control input choice. Of all the control input possibilities, there always exists one such that the robot will be able to stop before hitting any obstacle to insure safety [17].

Primbs, Nevisti, and Doyle explore Control Lyapunov Functions (CLF) and MPC approaches to nonlinear control [18]. Comparisons are made between the two methods and it is noted that they seek to solve the same problem optimally and

some properties of each method are complementary to one another. The CLF methods are best interpreted in the context of Hamilton-Jacobi-Bellman equations while MPC relates more closely to a Euler-Lagrange framework. The CLF provides a global optimal solution, but the partial differential equation one must solve is very difficult and computationally infeasible. The MPC approach instead allows for on-line computation of an optimal solution locally, and resolves this problem every time step specified. The problem is only solved over a fixed time horizon. However, it is difficult to apply Lyapunov Stability Theory to this MPC method where numerical techniques are used to solve for optimal solutions. Also, the relation between time horizon length and stability are not necessarily linear as noted by results from experiments [18].

Tahirovic and Magnani proposed a passivity based MPC control for Robot Navigation through rough terrain [19]. This method boasts simplicity of application for all vehicles as long as one can accurately model them. A virtual model of the vehicle is then made using shaped energy. The algorithm quantifies roughness of the terrain and uses this parameter when analyzing the cost of a specified path. The roughness is expressed as the relative height of the terrain locally. Simulations have proven successful navigation of a vehicle through hilly terrain, yet there is no mention of how the terrain data would actually be obtained [19]. This passivity based MPC method can be applied on-line, but this would only be feasible if there was some sort of sensor and data analysis algorithm which can sense the local terrain roughness.

Bevan, Gollee, and O'Reilly proposed a new method for trajectory generation for road vehicle obstacle avoidance using convex optimization [20]. The dynamics of a vehicle is a very nonlinear case which on the surface poses a non-convex optimization problem when searching for an optimal path trajectory. This study used their method on a vehicle performing an aggressive double lane change

maneuver with the intent of designing a quick emergency obstacle avoidance algorithm for when cars either need to stop immediately or turn to move around an incoming obstacle. Most avoidance systems solely slow the vehicle down to prevent collision with a hazard in the vehicle heading direction. The optimization is performed in three stages in which each stage includes different assumptions to frame the problem as a convex optimization problem. The first stage solves for a trajectory assuming no slip, the second assumes slip but constant speed, and the third uses the values from the previous stages to insert non-convex expressions and holds those terms constant. This study successfully simulated a general wheeled vehicle performing an optimal double lane change maneuver [20].

Nanao and Ohtsuka presented a nonlinear MPC algorithm which finds a solution using continuation/generalized minimal residual (C/GMRES) algorithm to find a solution [21]. The algorithm was tested successfully in simulation and though runtimes are not yet quick enough for real-time implementation, they expect it in the near future of this paper. Tire ground interactions are modeled using the Pacejka Magic Formula. A friction circle is used to quickly determine the maximum friction force able to be generated and this is used to create an “unavoidable region” in which the vehicle is unable to avoid hitting the obstacle [21]. However, this method relies heavily on accurate modeling of the ground tire interactions.

Gao, et al. presented and tested two different MPC algorithms for obstacle avoidance [22]. The first algorithm is a one level MPC algorithm. The algorithm combines the requirements of obstacle avoidance and the optimal trajectory planning into one step, solving a highly nonlinear optimal control problem within the horizon each time step. The objective function includes a proximity term which increases as the vehicle approaches an obstacle. The second algorithm is a two-level approach. A high-level path planner calculates an optimal trajectory for the vehicle disregarding any obstacle information. A low level algorithm then calculates the

optimal control inputs that both route the vehicle around obstacles and maintain the vehicle along the desired trajectory. While both algorithms were implemented in real-time successfully on an icy road performing a double lane change maneuver, the two-level approach was successful at speeds of 55 kph while the one level could only be implemented up to 40 kph. Computational runtimes were compared and show the two-level approach runs much quicker than the one-level approach and shows promise for hierarchical MPC methods in the future [22].

The two-level MPC method was developed further and addressed the issue of the vehicle model predicting infeasible trajectories for the vehicle in the high-level planner [23]. Instead, motion primitives are used to develop a high-level trajectory assembled from trims which are connected by maneuvers. Possible trims are straight, left turn, right turn, drift left, and drift right. Maneuvers connect two trims. The possible maneuvers include straight to left turn, left turn to straight, straight to drift right, etc. This method proved successful in simulation. In experiments, a vehicle was able to successfully avoid obstacles, but no mention is made of the vehicle speed [23].

### 1.2.3 MPC Algorithms for Car-Like Vehicle

Several studies have been performed examining MPC-based techniques for car-like vehicle collision avoidance. A MPC algorithm was developed that commands the braking and steering inputs to better control the vehicle during a double lane change maneuver [24]. This algorithm was tested on a snowy road. In experiments, a controller that controlled both braking and steering performed better than an algorithm commanding steering alone [24].

Abbas, Eklund, and Milman analyzed the possible real-time implementation of a nonlinear MPC algorithm for obstacle avoidance using Simulink and simulating the vehicle as a full Carsim nonlinear multibody vehicle [25]. Offline a trajectory is generated guiding the vehicle from a start point to the desired end location. Then

on-line each controller time step a cost function is used to weigh possible trajectories and deviations from the reference trajectory are penalized. A pointwise potential function is used to increase cost as the vehicle gets closer to any obstacle. Simulation results showed the vehicle is capable of navigating around a single obstacle. The cost function is minimized using the gradient descent optimization method. The amount of time to find an optimal path in the simulation varies depending on the scenario the vehicle is in at each time step. A vehicle in close proximity to an obstacle will take longer to pick the best path than a vehicle in free space pointed towards the goal. Warm starting is used during the optimization routine to help quicken the optimization process, but there are still many time step scenarios where the amount of time needed to calculate a best path exceeds the available time the controller has to find that path [25].

Researchers have proposed a fast motion planner that uses a half-car dynamical model for a wheeled vehicle [26]. A fast local steering algorithm was developed to improve runtimes. The algorithm splits into a geometric path planning step and then an optimal time parameterization step. The three control inputs are steering angle and the longitudinal slips of the front and rear tires much like a rally car driver would control the vehicle. Simulation results support the real-time implementations of this obstacle avoidance controller [26].

Gray, et al. used a MPC algorithm to maintain driver safety by keeping a vehicle within its lane [27]. The controller has an internal four wheel vehicle model as well as a driver model, allowing the controller to predict driver commands and intentions. The controller aims to minimize its own input so that only when the driver is in danger such as departing from the lane will the controller take action. Otherwise, it seeks to maintain safety constraints so that the controller only intervenes when it senses safety constraints may be violated. Simulation results proved successful, but when implemented in real life at a Volvo facility, the only issue encountered was

minor breaking of a safety constraint. This safety violation was due to sensor delay and in the future this delay will be accounted for in the algorithm [27].

Beal and Gerdes proposed a solution to the issue of vehicle stabilization control near the limits of handling [28]. Normally, an Electronic Stability Control (ESC) system works by using a linear model of a vehicle to predict driver intent and when the vehicle deviates past a threshold of the allowed deviation between driver intent and vehicle behavior, then the controller activates and stabilizes the vehicle.

However, this method does not work when the vehicle is being operated near the stable vehicle limits such as by an experienced driver because the linearized vehicle model is linearized about a conservative point for production level cars. The vehicle behavior near its dynamic limits is nonlinear and does not match the linearized model. Therefore, an MPC based controller was proposed and tested that operates at two levels and quickly enough for real-time implementation. The controller has two objectives. The first is to keep the vehicle inside of the safe-handling envelope and respond appropriately in case the vehicle leaves this safe envelope. The second objective is to allow the controller to track the driver's intended trajectory. Overall the controller was able to successfully achieve these objectives through experiments with Stanford's PI vehicle testbed [28].

Ersal, et al. developed a constant speed LIDAR-based local obstacle avoidance controller using an MPC algorithm and performed a study on the role of model fidelity with this controller [5]. The internal vehicle models used for this study were 2-DOF and 14-DOF constant speed vehicle models while lateral tire forces were modeled either using the pure slip Pacejka Magic Formula or a linearized version of the Pacejka Magic Formula. All of these tests were performed on rigid ground and a 14-DOF vehicle model was used as the plant vehicle. The vehicle when controlled by the controller with the 14-DOF model with nonlinear tire forces performed marginally better in this study than all other vehicle and tire force model

combinations, but runtimes were sacrificed to achieve this boost of performance. However, the 2-DOF models with linear tire forces did perform successfully in most tests which led to the conclusion that using this controller would be appropriate due to the benefits of quicker runtimes. The 14-DOF model can then be used to determine appropriate safety constraints to include in the optimal control problem formulation [5].

Ersal, et al. also developed a combined speed and steering local obstacle avoidance controller using an MPC algorithm which is an improvement on the controller from [5]. A multi-phase optimal control problem was formulated to optimize speed and steering within the prediction window [4]. A powertrain model is used to determine acceleration capabilities as a function of speed of the vehicle and safety constraints to prevent tire lift-off are imposed in the optimal control problem formulation. The vehicle does not need *a priori* knowledge of the obstacle field and instead develops a safe area polygon from LIDAR points. Simulation results from this study show this newer controller is capable of navigating through more obstacle distributions and sharper turns than the constant speed controller. A 3-DOF vehicle model is used as the internal controller model used to predict vehicle states [4].

#### 1.2.4 Extending Rigid Ground Tests to Granular Terrain

A common aspect to all of the previously mentioned papers and studies is that they were performed on rigid ground. Some sources used height maps to test the vehicle over hilly terrain or used a roughness factor to quantify any deviations from flat terrain. However, most of the previous studies had been primarily focused on the case where a vehicle needs to move to a target location on rigid terrain. Depending on the application, this sort of testing may be appropriate such as for a car driving on a paved road. For applications such as military, agricultural, off-road sport, and mining this rigid ground assumption does not hold. Therefore a fundamental understanding of the relationship between vehicle performance and

terrain characteristics is necessary. The testing performed previously in [5] should be extended to granular terrain. Extending the experimentation and simulation to granular terrain leads to another two questions. What is the best way to simulate granular terrain and what effect do the granular terrain parameters have on vehicle performance?

### 1.2.5 Simulating Granular Terrain for Vehicular Applications

The first question depends on the application and the computational resources available for the study. Some methods are quicker than others yet do not completely capture exactly what is happening at the tire ground contact patch while other methods more accurately model that interaction but are computationally exhausting. Generally in terramechanics there are two high fidelity techniques to model soft soils for a vehicle on granular terrain applications: finite element models and particle or mesh-free models.

Finite Element (FE) soil models model the soil by discretizing it into a FE mesh and use some elasto-visco-plastic continuum mechanics constitutive material model to approximate the behavior of the soil such as yielding, flow, cohesion, and internal friction. A review of basic soil constitutive models was compiled in [29]. These models aim to approximate the Mohr-Coulomb yield behavior of the soil. Most FE soil studies model the soil deformation using the motion of the FE nodes using a Lagrangian formulation. However, the Lagrangian method requires remeshing when deformations approach moderate to large values which are both computationally expensive and still may not prevent element distortion. The deformation of the FE mesh also fails to capture effects of soil deformation such as bulldozing or soil separation. High fidelity FE simulation of soil to include bulldozing and soil separation is an active area of research where there is even an effort to develop a soil continuum mechanics model which includes cohesion and can properly account for soil flow, plasticity, inter-particle friction, cohesion, and the effect of current stress

values on the material properties.

The high fidelity alternative to FE soil models is mesh-free particle-based models where instead of treating the terrain as a continuum, the terrain consists of a set of particles with inter-particle forces then representing the soil mechanical behavior. A brief summary of three different particle-based methods is presented in [30]. These methods are successfully able to capture the effects of soil flow, bulldozing, and ruts. Though particle-based methods are computationally expensive since typically they can have millions of particles, the benefits of handling bulldozing, separation, flow, and ruts make it a desirable method. They may even be less expensive than an FE method which is constantly being remeshed. One common particle-based method is the Discrete Element Method (DEM) where the soil is modeled as a large number of individual bodies which collide and roll along each other to simulate bulk soil material behavior. Normal contact forces generally in this method prevent particles from penetrating one another. Often DEM soil models include tangential contact forces, cohesive forces such as if the soil had some water content, or even distance dependent forces such as gravitational or electrostatic interactive forces.

Jayakumar, et al. performed a simulation study to understand the effects of soil characteristics on vehicle mobility over granular terrain [30]. Different cohesive and non-cohesive soils were modeled using DEM. Cone penetrometer simulations were performed to study the effect of the inter-particle friction coefficient and cohesion on soil strength measured using the cone index (CI). A multibody dynamics model of a vehicle was developed consisting of 33 rigid bodies connected with spherical, revolute, prismatic, and constant velocity joints. To simulate the vehicle over granular terrain, a moving soil patch was used which takes particles behind the vehicle, deletes them, and then re-emits them in front of the vehicle and compresses them before the vehicle moves over them again. This patch is implemented in the x-direction. This research group was able to conclude from their results that for

non-cohesive or weakly cohesive particles when particle size increases, wheel slip and sinkage decreases, vehicle speed increases, and tractive force slightly decreases. As soil cohesion increases, soil cone index increases resulting in vehicle speed increasing, wheel slip and sinkage decreasing, and tractive force decreasing. For weakly cohesive soil, inter-particle friction and soil density have a big influence on mobility parameters. Mobility parameters seem to level off once inter-particle friction reaches a certain point [30].

### 1.2.6 Summary

The performed literature review has assisted with refining the goals and means for achieving the goals of this research. Researchers for decades have been exploring obstacle avoidance techniques to navigate vehicles operating at their dynamic limits. However, the next logical step forward from this point is to implement these control algorithms on granular terrain. This task will be accomplished virtually by creating a test environment to perform both rigid ground and granular simulations using Chrono.

## 1.3 Objectives and Methodology

From extensive literature review, the goal of this research is to first understand how model fidelity of the controller model affects overall performance of the obstacle avoidance controller so that an improved controller can be developed to navigate a vehicle on granular terrain. The initial controller considered is a MPC LIDAR-based constant speed local obstacle avoidance controller presented in [5]. Identifying the role of model fidelity leads to a better understanding of vehicle dynamics and performance on granular terrain as well as promotes the development of a controller designed for general terrain travel in the future. From this understanding, the controller and internal models were improved to allow for better controlled vehicle performance on granular terrain. Simulating a vehicle on granular terrain in such

scenarios introduces additional challenges related to the sheer number of necessary particles required to properly model the desired terrain patch. This challenge has been addressed by employing a simulation framework for granular terrain described later in this thesis.

The present study has the following five objectives:

1. Develop a robust simulation environment using **Chrono** to simulate granular terrain efficiently over large terrains without actually modeling every single particle over the large terrain.
2. Study the performance of the MPC Controller on granular terrain compared to that on rigid terrain.
3. Analyze the impact of fidelity of the internal controller model on speed and performance of the obstacle avoidance controller.
4. Improve the controller for better performance on granular terrain.
5. Showcase the potential of controller testing in a high fidelity virtual test environment with **Chrono** to assist with initial control algorithm development before physical implementation for vehicular applications.

From previous literature research, this MPC algorithm has been tested successfully on flat rigid terrain. However, using **Chrono::Parallel** one has the capabilities to develop a simulation of a vehicle on hilly, deformable, and even granular terrain. In this research, **Chrono::Parallel** is used to develop a simulation of an AGV driving on customizable granular terrain through a user defined obstacle field to a defined target location. **Chrono** already possesses the ability to create new geometries easily with a modular framework, so one will then be able to easily create their own obstacle field for the vehicle to maneuver through. Using theory and developments from [5], the MPC algorithm has been implemented in the

simulation to determine optimal steering commands for the AGV. Tests similar to those executed in [5] were conducted within this simulation on rigid flat terrain for the two models presented previously to confirm the controller operates appropriately. The algorithm was then tested on granular terrain to understand how controlled vehicle performance changes on granular terrain versus previously tested rigid ground simulations. The MPC algorithm solves a very large optimization problem to determine the quickest safe path to the destination. An exhaustive search based method was used to solve this optimal control problem. Improvements were explored to boost vehicle performance on granular terrain. A second virtual experiment was conducted using the improvements to the controller and those results are discussed at the end of this thesis.

## 2 THEORETICAL BACKGROUND

This chapter provides an overview of theoretical topics relevant to this research. To accomplish the objectives of this study, a virtual simulation environment was developed in C++ using **Chrono**, the multibody physics API, to handle simulation of multibody systems and contact force calculation using the Penalty Method. The MPC LIDAR-based constant speed local obstacle avoidance controller used for this research is also outlined as well as the internal vehicle models used by the controller to predict future vehicle trajectories. The developed methods of implementing a perfect planar LIDAR sensor, the formulation of an optimal control problem, and the exhaustive search space method used in this research are presented as well.

### 2.1 Multibody Dynamics Background

Multibody Dynamics allows one to use a system of equations to represent the kinematic and dynamic behavior of a system of bodies. This section outlines the formulation of multibody dynamics implemented in **Chrono**. A system of index 3 Differential Algebraic Equations (DAE's) are used to characterize the dynamics of a system composed of rigid and flexible bodies as presented in [1].

$$\dot{\mathbf{q}} = \mathbf{L}(\mathbf{q}) \mathbf{v} \quad (2.1)$$

$$\mathbf{M}(\mathbf{q}) \dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}) - \mathbf{g}_q^T(\mathbf{q}, t) \hat{\lambda} + \sum_{i \in \mathcal{A}(\mathbf{q}, \delta)} (\hat{\gamma}_{i,n} \mathbf{D}_{i,n} + \hat{\gamma}_{i,u} \mathbf{D}_{i,u} + \hat{\gamma}_{i,w} \mathbf{D}_{i,w}) \quad (2.2)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{q}, t) \quad (2.3)$$

$$i \in \mathcal{A}(\mathbf{q}(t)) : \begin{cases} 0 \leq \Phi_i(\mathbf{q}) \perp \hat{\gamma}_{i,n} \geq 0 \\ (\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) = \underset{\sqrt{\bar{\gamma}_{i,u}^2 + \bar{\gamma}_{i,w}^2} \leq \mu_i \hat{\gamma}_{i,n}}{\operatorname{argmin}} \mathbf{v}^T \cdot (\bar{\gamma}_{i,u} \mathbf{D}_{i,u} + \bar{\gamma}_{i,w} \mathbf{D}_{i,w}) \end{cases} \quad (2.4)$$

Equation (2.1) relates the time derivative of the generalized coordinates  $\mathbf{q}$  to the velocities  $\mathbf{v}$  using a linear transformation defined by  $\mathbf{L}(\mathbf{q})$ . Equation (2.2) relates the inertial forces of the system to the applied forces  $\mathbf{f}(t, \mathbf{q}, \mathbf{v})$  and constraint forces  $-\mathbf{g}_q^T(\mathbf{q}, t) \hat{\lambda}$ .  $\hat{\lambda}$  appears due to the constraints restricting the kinematic behavior of joined bodies in the system. Equation (2.3) is a set of bilateral constraints imposed on the system that enforce the kinematic behavior of all the bodies in the system and restrict their relative motion. The bilateral constraints of Equation (2.3) can then be augmented by unilateral constraints due to contact/impact phenomena. This point allows the concept of equations of motion to be extended to include differential inclusions [31]. A simple example that outlines this concept is of a single body that interacts with the ground through both friction and contact and the equations of motion become an inclusion  $\mathbf{M}\ddot{\mathbf{q}} - \mathbf{f} \in \mathbf{F}(\mathbf{q}, t)$ . In this case  $\mathbf{M}$  is the inertia matrix of the body,  $\ddot{\mathbf{q}}$  is the body acceleration,  $\mathbf{f}$  is the external force, and  $\mathbf{F}(\mathbf{q}, t)$  is some set-valued function. This inclusion then states that the frictional contact force remains somewhere inside of a friction cone with a not yet determined value controlled by the stick or slip state of the interaction between the body and the ground. The red terms in Equations (2.1-2.4) are introduced when in multibody dynamics the differential inclusion is posed as a differential variational inequality problem. The unilateral constraints then presented in Equation (2.4) define a set of contact complementarity conditions  $0 \leq \Phi_i(\mathbf{q}) \perp \hat{\gamma}_{i,n} \geq 0$ . These conditions make the following simple point regarding contact forces. For a potential contact  $i$  in the active set,  $i \in \mathcal{A}(\mathbf{q}(t))$ , either the gap  $\Phi_i$  between two geometries is zero and therefore the normal contact force  $\hat{\gamma}_{i,n}$  is greater than zero, or  $\Phi_i$  is greater than zero meaning two geometries are not in contact and therefore  $\hat{\gamma}_{i,n}$  is zero. Equation (2.4) then poses an optimization problem whose first order Karush-Kuhn-Tucker optimality conditions are equivalent to the Coulomb dry friction contact model [32]. Therefore in Equation (2.2) the force associated with contact  $i$  leads to the red

generalized forces obtained by using projectors  $\mathbf{D}_{i,n}$ ,  $\mathbf{D}_{i,u}$ , and  $\mathbf{D}_{i,w}$ .

## 2.2 Dealing with Contacts: The Penalty Method

The method of modeling the equations described above is implemented in **Chrono** to analyze the dynamics of large multibody systems and granular media in a DEM framework. This method uses the complementarity conditions in Equation (2.4) to enforce non-penetration of bodies that come into contact. This method is then referred to as DEM-C. This research however does not use the DEM-C method, but instead employs a Penalty (P) method that accounts for partial body deformations in mutual contact to calculate contact forces and thus this method is referred to as DEM-P. Therefore the Penalty Method is used to calculate contact forces one body exerts on another within the DEM simulation framework in **Chrono** chosen for this research [1]. With this DEM-P approach or soft-body approach, bodies are considered “soft” in that their collision geometries are able to overlap each timestep or experience local deformation. A collision detection algorithm is used to find when and where these overlaps  $\delta_n$  occur. Contact force vectors  $\mathbf{F}_n$  and  $\mathbf{F}_t$  normal and tangential to the contact plane respectively at the point of contact are calculated using a chosen constitutive law based on the local body deformations at the point of mutual contact. In the contact-normal direction  $\mathbf{n}$ , this local body deformation is defined as the overlap of the two quasi-rigid bodies  $\mathbf{u}_n = \delta_n \mathbf{n}$ . Then in the contact-tangential direction, the deformation is defined by the vector  $\mathbf{u}_t$  that keeps track of the total tangential displacement of the initial contact points on the two quasi-rigid bodies while in contact, projected onto the current contact plane.

An example of a DEM-P contact constitutive law similar to one implemented in **Chrono** is a viscoelastic model based on either Hookean or Hertzian contact theory outlined in Equations (2.5) and (2.6). Here  $\mathbf{u} = \mathbf{u}_n + \mathbf{u}_t$  is the overlap or local contact displacement of two interacting bodies. The quantities

$\bar{m} = m_i m_j / (m_i + m_j)$  and  $\bar{R} = R_i R_j / (R_i + R_j)$  are the effective mass and effective

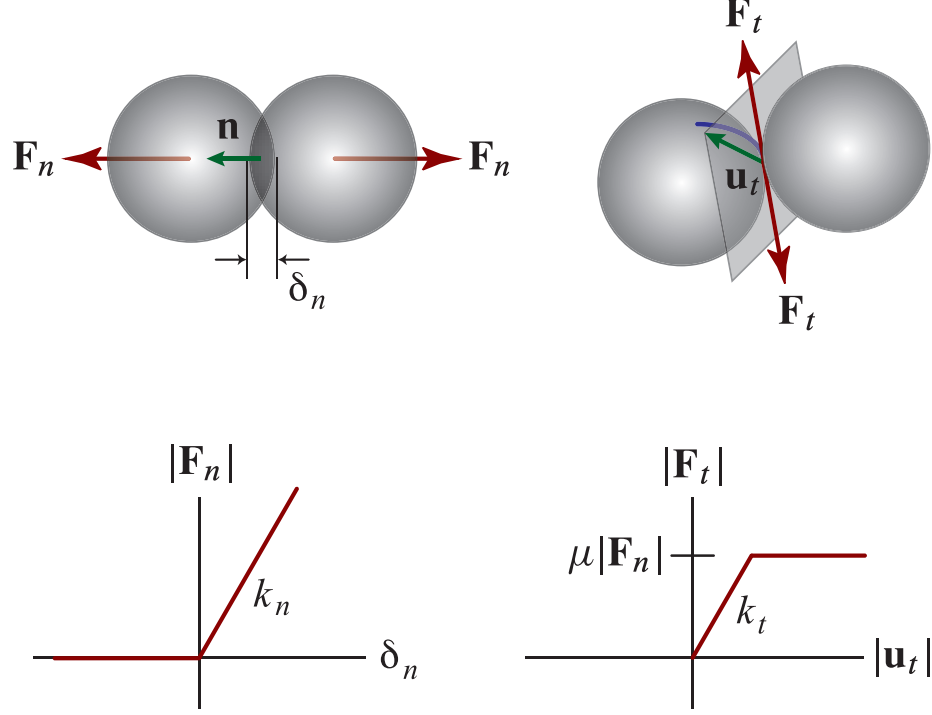


Figure 2.1: Reproduced from [1]. DEM-P contact model with normal overlap distance  $\delta_n$ , contact-normal unit vector  $\mathbf{n}$ , and tangential displacement vector  $\mathbf{u}_t$  (top), and a Hookean-linear contact force-displacement model with constant Coulomb sliding friction (bottom)

radius of curvature respectively for bodies  $i$  and  $j$  in contact. The vectors  $\mathbf{v}_n$  and  $\mathbf{v}_t$  are the normal and tangential components respectively of the relative velocity at the contact point. For a Hookean contact model,  $f(\bar{R}, \delta_n) = 1$  while for a Hertzian contact model,  $f(\bar{R}, \delta_n) = \sqrt{\bar{R}\delta_n}$  [33, 34, 35]. The normal and tangential stiffnesses and damping coefficients,  $k_n$ ,  $k_t$ ,  $\gamma_n$ , and  $\gamma_t$ , respectively are obtained through various constitutive laws derived from contact mechanics and physically measurable properties of materials of the contacting bodies such as Young's Modulus, Poisson's ratio, coefficient of restitution, etc.

$$\mathbf{F}_n = f(\bar{R}, \delta_n) (k_n \mathbf{u}_n - \gamma_n \bar{m} \mathbf{v}_n) \quad (2.5)$$

$$\mathbf{F}_t = f(\bar{R}, \delta_n) (-k_t \mathbf{u}_t - \gamma_t \bar{m} \mathbf{v}_t) \quad (2.6)$$

The component of the contact displacement vector  $\mathbf{u}$  in the contact normal

direction  $\mathbf{u}_n = \delta_n \mathbf{n}$  is obtained directly from the contact detection algorithm which identifies and quantifies the magnitude of body overlap between interacting bodies.

The tangential contact displacement vector  $\mathbf{u}_t$  is a bit more complicated and formulated in Equation (2.7). In Equation (2.7),  $t$  is the current time and  $t_0$  is the time when the contact was initiated. Then for a true tangential contact displacement history model, the vector  $\mathbf{u}_t$  must be stored and updated at each time for each contact point from the time when contact is initiated up until the contact is terminated. A visualization of the Penalty method is presented in Figure 2.1.

$$\mathbf{u}_t = \int_{t_0}^t \mathbf{v}_t dt - \left( \mathbf{n} \cdot \int_{t_0}^t \mathbf{v}_t dt \right) \mathbf{n} \quad (2.7)$$

To enforce the Coulomb friction law, if  $|\mathbf{F}_t| > \mu |\mathbf{F}_n|$  at any given time step, before the contact forces are added to the resultant force and torque on each body, the stored value of  $|\mathbf{u}_t|$  is scaled such that  $|\mathbf{F}_t| = \mu |\mathbf{F}_n|$ , where  $\mu$  is the Coulomb friction coefficient. An example of how  $|\mathbf{u}_t|$  is scaled is presented in Equation (2.8) if a Hookean contact model is used and  $f(\bar{R}, \delta_n) = 1$  in Equations (2.5) and (2.6).

$$k_t |\mathbf{u}_t| > \mu |\mathbf{F}_n| \Rightarrow \mathbf{u}_t \leftarrow \mathbf{u}_t \frac{\mu |\mathbf{F}_n|}{k_t |\mathbf{u}_t|} \quad (2.8)$$

Through the processes described above, once the contact forces  $\mathbf{F}_n$  and  $\mathbf{F}_t$  are computed for each contact and their contributions are summed to obtain resultant forces and torques for each interacting body, the Newton-Euler equations of motion are integrated subject to the Courant-Friedrichs-Lewy (CFL) stability condition.

The CFL stability condition limits the integration time step-size  $h$  to

$h < h_{crit} \sim \sqrt{m_{min}/k_{max}}$  [36]. Here  $h_{crit}$  is the critical integration step size  $h$  must

stay below to maintain simulation stability,  $m_{min}$  is the smallest mass in the

simulation, and  $k_{max}$  is the largest stiffness in the simulation. For multibody

dynamics simulations with frictional contact modeled using the penalty approach as

with this research, **Chrono** implements index 3 DAE solutions [1].

## 2.3 Chrono Multibody Physics Package

The physics modeling and simulation capabilities are provided by the multiphysics open-source package **Chrono** [1]. The core functionality of **Chrono** provides support for the modeling, simulation, and visualization of rigid multibody systems, with additional capabilities offered through optional modules. These modules provide support for additional classes of problems (e.g., finite element analysis and fluid-solid interaction), for modeling and simulation of specialized systems (such as ground vehicles and granular dynamics problems), or provide specialized parallel computing algorithms (multi-core, GPU, and distributed) for large-scale simulations.

### 2.3.1 Vehicle Modeling

Built as a **Chrono** extension module, **Chrono::Vehicle** [37] is a C++ middleware library focused on the modeling, simulation, and visualization of ground vehicles. **Chrono::Vehicle** provides a collection of templates for various topologies of both wheeled and tracked vehicle subsystems, as well as support for modeling of rigid, flexible, and granular terrain, support for closed-loop and interactive driver models, and run-time and off-line visualization of simulation results.

Modeling of vehicle systems is done in a modular fashion, with a vehicle defined as an assembly of instances of various subsystems (suspension, steering, driveline, etc.). Flexibility in modeling is provided by adopting a template-based design. In **Chrono::Vehicle**, templates are parameterized models that define a particular implementation of a vehicle subsystem. As such, a template defines the basic modeling elements (bodies, joints, force elements), imposes the subsystem topology, prescribes the design parameters, and implements the common functionality for a given type of subsystem (e.g., suspension) particularized to a specific template (e.g.,

double wishbone). Finally, an instantiation of such a template is obtained by specifying the template parameters (hardpoints, joint directions, inertial properties, contact material properties, etc.) for a concrete vehicle (e.g., the HMMWV front suspension).

For wheeled vehicle systems, templates are provided for the following subsystems: *suspension* (double wishbone, reduced double wishbone using distance constraints, multi-link, solid-axle, McPhearson strut, semi-trailing arm); *steering* (Pitman arm, rack-and-pinion); *driveline* (2WD and 4WD shaft-based using specialized **Chrono** modeling elements, simplified kinematic driveline); *wheel* (simply a carrier for additional mass and inertia appended to the suspension’s spindle body); *brake* (simple model using a constant torque modulated by the driver braking input).

In addition, **Chrono::Vehicle** offers a variety of tire models and associated templates, ranging from rigid tires, to empirical and semi-empirical models (such as Pacejka and Fiala), to fully deformable tires modeled with finite elements (using either an Absolute Nodal Coordinate Formulation or a co-rotational formulation). Driver inputs (steering, throttle, and braking) are provided from a *driver* subsystem with available options in **ChronoVehicle** including both open-loop (interactive or data-driven) and closed-loop (e.g., path-following based on PID controllers).

#### 2.4 MPC LIDAR-Based Local Obstacle Avoidance: History and Current Algorithm

The concept of MPC is to use an internal model of the system one desires to control to predict and optimize future system behavior from the current system state and inputs [16]. The system behavior is predicted over some defined finite time horizon and the optimal control sequence over the prediction horizon is output. The control sequence is executed for an execution time smaller than the prediction horizon, and the whole process is repeated. The repetition of this process over time creates a feedback loop which continually controls the system, pushing it towards an optimal path.

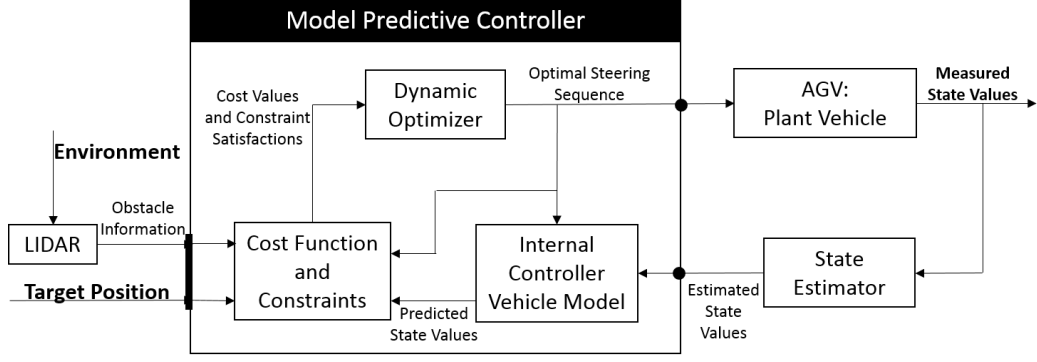


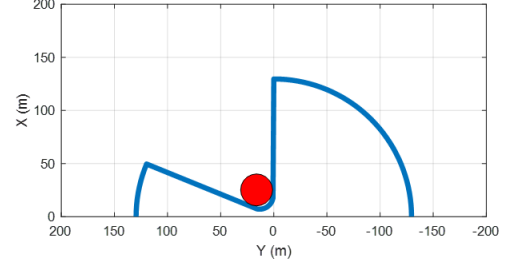
Figure 2.2: Schematic of MPC LIDAR-Based Constant Speed Local Obstacle Avoidance Controller

For this study, the system to be controlled is an AGV. Consider an AGV located in a level environment without roads or any other structures to guide its motion and assume the AGV has a known global target position. Between the target position and the current vehicle position there may or may not be obstacles of unknown size. Using the MPC formulation outlined in [5], the vehicle can navigate from the current position to the provided target position while avoiding obstacles as they are encountered. Obstacle information is assumed to be unknown *a priori* and only obtained through a planar LIDAR sensor. The MPC schematic is presented in Figure 2.2.

The planar LIDAR sensor, mounted at the front center location of the vehicle, returns the closest obstacle boundary in all radial directions of the sensor at an angular resolution  $\epsilon$ . The sensor has a maximum range past which it cannot sense any obstacles. Therefore, if the closest obstacle boundary is greater than the LIDAR radius  $R_{LIDAR}$ , then the sensor returns  $R_{LIDAR}$ . The LIDAR sensor range is  $[0^\circ, 180^\circ]$ , with  $90^\circ$  being the vehicle heading direction. Since the AGV considered here is driving along level ground, whether granular or rigid, a planar LIDAR sensor is sufficient. The sensor is assumed to have no delay and zero noise and can therefore instantaneously generate a safe area polygon assembled from the returned points from the LIDAR. An overhead view of the AGV encountering an obstacle



(a) 3D Visualization of LIDAR Encountering Obstacle



(b) LIDAR Sensed Safe Area

Figure 2.3: Sample obstacle field and LIDAR output

and the generated LIDAR safe area polygon are presented in Figure 2.3.

In this research, all of the obstacles are upright cylinders so a ray-circle intersection algorithm is used to determine when each LIDAR ray intersects with a nearby obstacle. The ray-circle intersection algorithm is presented in Appendix A. This algorithm does a quick and efficient job of simulating a perfect LIDAR sensor encountering upright cylinders when the algorithm is performed on a set of rays all sweeping iteratively from the vehicle's location.

For the purpose of these simulations, the only outputs of the MPC algorithm are the steering signals since the plant vehicle maintains a near constant speed controlled by a PID speed controller. As shown in Figure 2.2, the MPC algorithm is made up of the internal controller vehicle model, the cost function and constraints, and the dynamic optimizer. The internal controller vehicle model predicts the future states of the AGV for a given steering sequence. Herein, the internal controller vehicle model is varied from test to test between a 2-DOF vehicle model and a 14-DOF vehicle model, as detailed in further sections. Cost functions and constraints are used to formulate the optimal control problem using the equations from the vehicle model, and the resulting optimal control problem is solved using dynamic optimization.

Since the ability of finding and executing an optimal solution, rather than

solution speed, is the primary focus here, an exhaustive search is used to find the optimal solution to the problem at hand. With this method, the steering sequence is discretized and a finite set of path possibilities are tested and weighed by a cost function.

The MPC controller as formulated in [5] is used for this study. The cost function and constraints need to be specified to avoid collisions with obstacles and guarantee vehicle dynamical safety. The optimal control problem solved at each MPC time step consists of the following set of equations:

$$\underset{J}{\text{minimize}} \quad J = s_T + wd \quad (2.9)$$

$$\text{subject to} \quad \dot{\xi} = v [\xi(t), \zeta(t)] \quad (2.10)$$

$$\xi(0) = \xi_0 \quad (2.11)$$

$$\tilde{S}[x(t), y(t)] \leq 0 \quad (2.12)$$

$$|\delta_f(t)| \leq \tilde{\delta}_{f,max}(U_0) \quad (2.13)$$

$$|\varsigma_f(t)| \leq \varsigma_{f,max} \quad (2.14)$$

$$t \in [0, T_P] . \quad (2.15)$$

Equation (2.9) defines the cost function for this optimal control problem where  $J$  is the cost of a path. This equation is a soft requirement which defines how the separate path possibilities are weighed against each other and how to determine which path is actually optimal. The cost function is comprised of two terms defined as:

$$s_T = \sqrt{[x_G - x(T_P)]^2 + [y_G - y(T_P)]^2} \quad (2.16)$$

$$d = \int_0^{T_G} |\varsigma(t)| dt . \quad (2.17)$$

Here,  $s_T$  seeks to minimize the distance between the prediction end-location  $(x(T_P), y(T_P))$  and the target position  $(x_G, y_G)$ . A prediction path that guides the vehicle towards the closest location to the target will have the smaller  $s_T$  term. The second term,  $d$ , aims to minimize the change in steering angle so that smoother, straighter paths are preferred over windy paths. A weighting factor  $w$  is used to scale the influence of  $d$  in the total cost.

Equations (2.10-2.15) are the constraints for this optimal control problem and represent hard requirements for vehicle safety and collision avoidance. Any paths that violate these constraints are not considered safe paths and are eliminated as potential options in this prediction window. Equation (2.10) defines a set of differential equations that describe the internal controller vehicle model where  $\xi$  is the set of system state variables and  $\zeta$  is the set of input command variables, with the initial conditions at the start of the time horizon set by Equation (2.11).

Equation (2.12) defines a safe area polygon, constructed from LIDAR data and including an additional safety buffer to account for vehicle size and to prevent collisions of the vehicle corners. All points along paths found from Equations (2.10) and (2.11) must fall inside this safe area polygon.

Equation (2.13) imposes a maximum limit on the front steering angle  $\delta_f$ , based on vehicle longitudinal speed. This value is obtained from a lookup table which can be generated either experimentally or from simulation. Equation (2.14) imposes a maximum limit on front steering rate  $\varsigma$ . Equation (2.15) defines the time horizon  $T_P$  over which this optimal control problem is solved.

To determine if any point along the path is within the LIDAR sensed safe area polygon, a point in polygon algorithm is used. The algorithm developed for this research is similar to the winding method where the winding method tracks how many times the edges of the polygon wind around the current point of interest. However, the algorithm developed and used in this research is much simpler and

quicker and is very similar to that provided in [38]. This algorithm checks the quadrant where each vertex of the polygon is located in a frame with origin at the current point of interest. The algorithm then counts the number of times that quadrant changes as the algorithm cycles through all of the vertices to determine if the point of interest is inside the polygon. The motivation for this actually comes from real life. Imagine sitting in a closed room. A way one could test if they were actually in the room would be to iteratively look at all the corners of the room in sequence. In the person's relative reference frame fixed in its orientation, cycling through the corners of the room one would note the net quadrant change is 4, regardless of how many corners there are. Now if the person were standing outside of the room and performed this same test, they would note that the net quadrant changes was a number other than 4 meaning the room did not enclose them and therefore they are outside of the room. Again the method is similar to the winding method but the algorithm performs and checks different aspects to reach the same conclusion quicker. This research uses a planar LIDAR sensor. Thus, this algorithm is only developed in two dimensions in the x-y plane. Further development of this point in polygon algorithm can be found in Appendix B.

## 2.5 Multi-DOF Vehicle Dynamics Models

Equation (2.10) refers to an analytical vehicle model expressed as a set of differential equations which allow the controller to predict vehicle performance within the prediction horizon. The accuracy of the internal vehicle controller model should directly influence the driven vehicle's controlled performance. If the internal vehicle controller model poorly predicts a vehicle's response to inputs from the driver or the environment, then these deficiencies will be witnessed when attempting to control the driven vehicle and result in incorrect trajectories that may lead to collisions. On the other hand, a highly complex internal vehicle controller model may provide accurate trajectory and vehicle response predictions, but with an

unacceptable time required to solve the optimal control problem. An ideal controller should not only accurately predict vehicle responses, but also do it quickly enough to insure vehicle safety. The vehicle models embedded in the MPC are simplifications of the full, multi-body based, **Chrono** wheeled vehicle model which is being controlled. Two models were considered initially, providing different levels of fidelity, as described below.

### 2.5.1 2-DOF Vehicle Model

The standard vehicle model used in recently developed MPC obstacle avoidance algorithms such as [5] is the 2-DOF yaw plane vehicle model. These models normally either assume constant cornering stiffness or the nonlinear Pacejka Magic Formula Tire Model [39] to predict the ground tire interaction forces. For this study, the Pacejka Magic Formula is used to predict tire forces in the vehicle models. This force model will be presented in Section 2.6.

A visual representation of the 2-DOF yaw plane vehicle model can be found in Figure 2.4. The 2-DOF model is described by the following first-order ordinary differential equations:

$$\dot{V} = (F_{y,f} + F_{y,r}) / M - U_0 r \quad (2.18)$$

$$\dot{r} = (F_{y,f} L_f - F_{y,r} L_r) / I_{zz} \quad (2.19)$$

$$\dot{\psi} = r \quad (2.20)$$

$$\dot{x} = U_0 \cos \psi - (V + L_f r) \sin \psi \quad (2.21)$$

$$\dot{y} = U_0 \sin \psi + (V + L_f r) \cos \psi, \quad (2.22)$$

where  $F_{y,f}$  and  $F_{y,r}$  are the lateral tire forces at the front and rear axles, respectively.  $U_0$  and  $V$  are the longitudinal speed and lateral speed of the vehicle in the vehicle's coordinate frame.  $\psi$  is the yaw angle and  $r$  is the yaw rate.  $(x, y)$

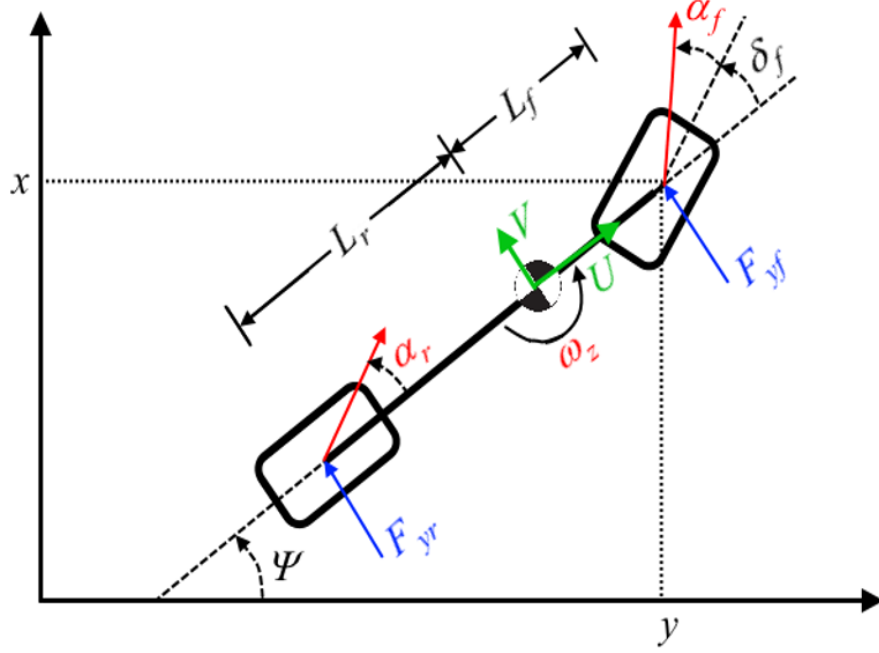


Figure 2.4: 2-DOF Vehicle Model

represent the front center location of the vehicle expressed in global coordinates.  $M$  is vehicle mass,  $I_{zz}$  is the moment of inertia of the vehicle about the CoG and the z-axis,  $L_f$  is the distance from the front axle to the vehicle CoG, and  $L_r$  is the distance from the rear axle to the vehicle CoG.

For this research, the model is constrained to a constant longitudinal speed. Then the planar 3-DOF body with one constraint results in the 2-DOF model used for this study.

### 2.5.2 14-DOF Vehicle Model

A 14-DOF model is often used in studies such as these to test the obstacle avoidance controller with a higher fidelity vehicle model [3, 5]. A benefit of using the 14-DOF in the controller is the model's ability to predict tire liftoff and account for dynamic effects from suspension systems. For this research, it is appropriate to also compare performance of the local obstacle avoidance controller running an internal 14-DOF on rigid terrain versus granular terrain.

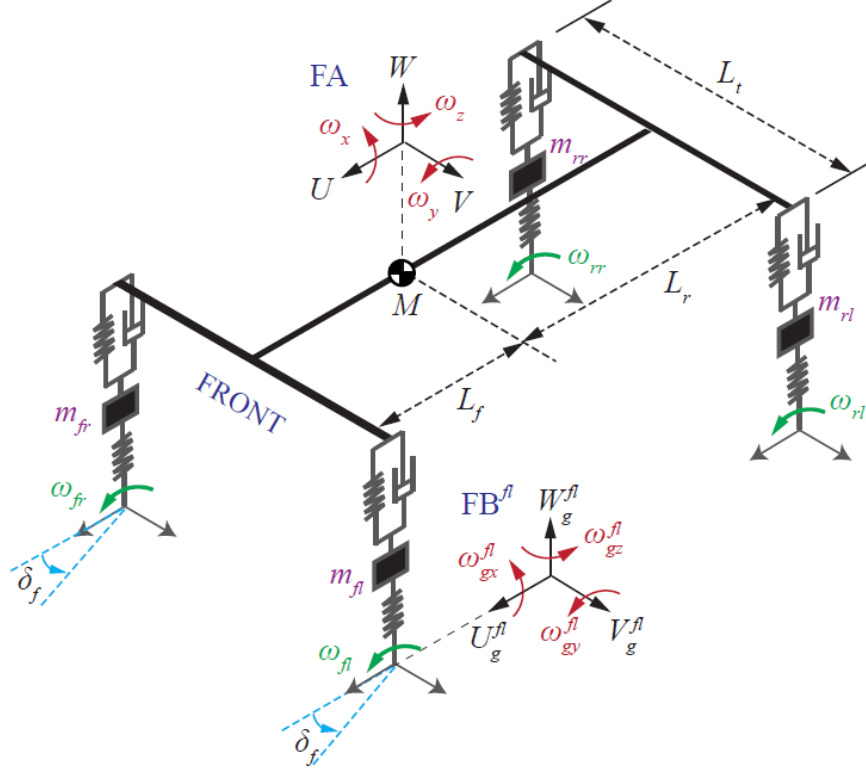


Figure 2.5: Reproduced from [2]. 14-DOF Vehicle Model

The 14-DOF vehicle model consists of one sprung mass connected above four unsprung masses [2]. The sprung mass is allowed to roll, pitch, and yaw while also displacing laterally, vertically, and longitudinally. This sprung mass contributes six DOF to the model. Each of the four wheels are allowed to bounce vertically and rotate about the wheel horizontal axis. The front two wheels are also free to steer. Each wheel then contributes two DOF to the 14-DOF model. The model is constrained at a constant longitudinal speed of 8.1 m/s used for this research. The equations used for this model, as well as their derivation can be found in [2] and visualized in Figure 2.5.

## 2.6 Tire Ground Force Model: Pacejka Magic Formula

The Pacejka Magic Formula is a semi-empirical tire model developed in 1987 by Hans B. Pacejka from Delft University of Technology, Egbert Bakker from Volvo

Cars, and Lars Nyborg from Volvo Cars [40]. This model was developed further by Pacejka and commercially known as the “MF-Tire.” The Magic Formula is capable of characterizing a tire with a large set of coefficients obtained from experiments. These coefficients are then used to calculate lateral, longitudinal forces as well as the self aligning torque experienced by a tire interacting with the ground.

$$A = B(F_z) \cdot \alpha \quad (2.23)$$

$$F_y = D(F_z) \cdot \sin \{C(F_z) \cdot \text{atan}[A - E(F_z) \cdot A + E(F_z) \cdot \text{atan}(A)]\} \quad (2.24)$$

The lateral forces at a wheel used in the previously described 2-DOF vehicle model are calculated using Equations (2.23) and (2.24) for the case of pure lateral slip.  $F_z$  is the vertical load at the axle.  $\alpha$  is the lateral tire slip angle.  $B(\cdot)$ ,  $C(\cdot)$ ,  $D(\cdot)$ , and  $E(\cdot)$  are functions of the tire vertical load  $F_z$  and these relationships are described in [39]. The slips at a front and rear tire are calculated using Equations (2.25) and (2.26) respectively where  $\delta_f$  is the steering angle of the front wheel.

$$\alpha_f = \tan^{-1} \left( \frac{V + L_f r}{U_0} - \delta_f \right) \quad (2.25)$$

$$\alpha_r = \tan^{-1} \left( \frac{V - L_r r}{U_0} \right) \quad (2.26)$$

The tire vertical load is then required as an input to Equations (2.23) and (2.24). Longitudinal load transfer is accounted for when calculating front and rear tire vertical loads. Justification for this inclusion can be found in [5]. Equations (2.27) and (2.28) are then used to calculate the front and rear tire loads  $F_{z,f}$  and  $F_{z,r}$  and input into Equations (2.23) and (2.24).

$$F_{z,f} = \frac{Mg L_r + MV r h_{CG}}{L_f + L_r} \quad (2.27)$$

$$F_{z,r} = \frac{MgL_f - MVrh_{CG}}{L_f + L_r} \quad (2.28)$$

The Magic Formula characterizes the tire's interaction with rigid ground with a set of coefficients and uses those coefficients to match the Magic Formula with experimental data. Then a tire interacting with one type of pavement will have different Magic Formula parameters than the same tire interacting with a different type of pavement. However, the Magic Formula models tire ground interactions on rigid ground, not granular terrain i.e., there is not a granular terrain Magic Formula. There is then a mismatch between the internal controller models and the plant vehicle when controlling the vehicle on granular terrain, but this research first seeks to identify what role that mismatch plays in controller performance. Improvements presented in Chapter 5 address this model mismatch in order to better account for the granular terrain interaction.

## 2.7 Optimization Problem: Exhaustive Search Space

When this controller is implemented in real-time on a physical vehicle, it will use some high efficiency optimization algorithm to compute which steering sequence is best for the HMMWV to follow. For this research, that is not the case because the interest lies in analyzing the role and effect of model fidelity on the controller performance. When the system dynamics are represented using the 14-DOF analytical vehicle model, the optimization problem becomes very difficult to solve and the techniques to perform such a complex optimal control problem quickly are still the subject of ongoing research. Considering this observation, an exhaustive search space approach is used for this research to solve the optimal control. This is the same technique used in [5].

Using this exhaustive search space method, the steering search space is discretized instead of considering the entire continuous steering search space. Some discrete steering angles are considered which include the maximum safe steering

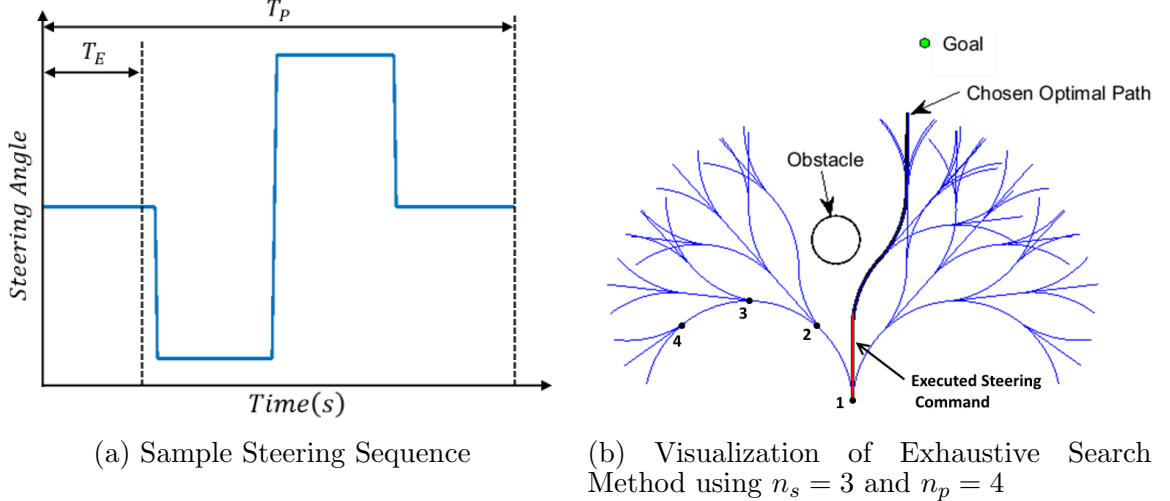


Figure 2.6: Exhaustive Search Space Simplified Visualization

angle, negative maximum steering angle, and zero steering. The set of discrete steering angles is then denoted as the steering angle pool, represented by Equation (2.29). The prediction horizon is divided into intervals of equal length. Then at each interval, a steering angle from the steering pool is selected and held constant for that interval. A zero-order hold approach is used to develop a steering angle sequence from the steering angle of each interval. Ramp transitions join two different steering angles so that the slope is the maximum steering rate. The prediction horizon is divided into  $n_p$  intervals and the number of steering angles considered in the steering pool is  $n_s$ . Therefore the total size of the search space or number of different possible steering sequences considered is  $n_s^{n_p}$ . The steering sequence is then executed over the execution horizon  $T_E$  which is set equal to or less than the interval length  $T_P/n_p$ . For this research, the parameters used are  $n_p = 4$ ,  $n_s = 5$ , and  $T_E = 0.5s$ . The discretization of the search space allows for a simplification of Equation (2.17) from the cost function which yields Equation (2.30).

$$\mathcal{O}_{steer} = [-\tilde{\delta}_{f,max}(U_0) \cdots 0 \cdots \tilde{\delta}_{f,max}(U_0)] \quad (2.29)$$

$$d = \sum_{i=1}^{n_p} |\delta_i - \delta_{i-1}| \quad (2.30)$$

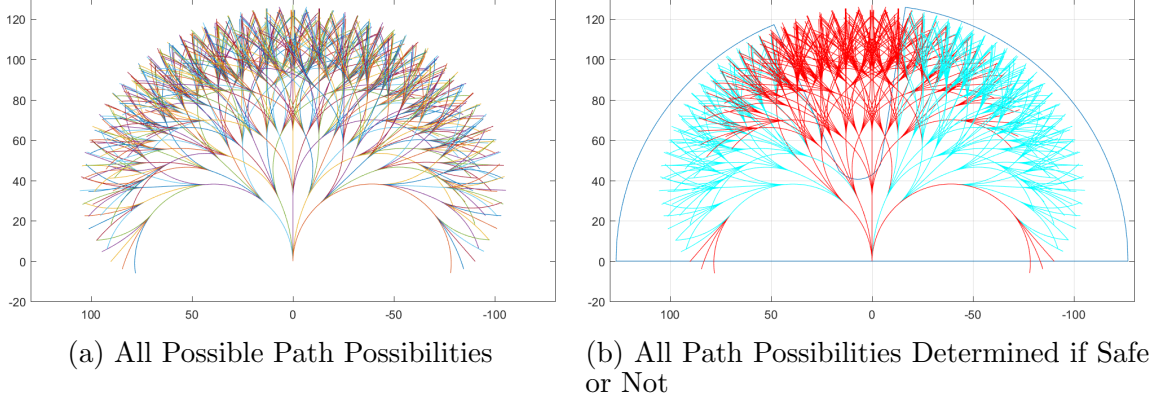


Figure 2.7: Exhaustive Search Space Paths Checked with Point in Polygon Algorithm

The control sequences are compared against each other using the cost function. The optimal control sequence that minimizes the cost function and satisfies the constraints of the optimal control problem is selected from the steering pool and executed over the execution horizon. Starting at the initial state, control commands from the steering pool are applied to the analytical vehicle model and the resulting trajectories are checked for constraint violation. If the constraints are violated, that search branch is terminated. Else, the next step prediction is performed by using the end state of the last analytical evaluation as the new initial state and applying commands from the steering pool. This process is repeated for all intervals and until the prediction horizon window has been reached. Once all control sequences have been evaluated, they are compared using the cost function. The control sequence that minimizes the cost function is considered the optimal control sequence. The optimal control sequence is executed over the execution window and the whole process is repeated. Refer to Figure 2.6 for an example implementation of this exhaustive search method. Figure 2.7 presents the full set of different path possibilities considered and how they are eliminated based on the LIDAR safe area polygon. For this research, the steering space is discretized into five steering angles, while the prediction horizon is split into four intervals, yielding 625 different steering sequence possibilities and therefore 625 different path possibilities that are

weighed by the controller to determine the optimal steering sequence.

### 3 EXPERIMENTAL SETUP

Based on the identified objectives of this research, a series of numerical simulations were setup to examine the role of model fidelity and the influence of terrain type on controlled vehicle performance. To accomplish this task, a robust simulation environment was developed in C++ that leverages the multibody capabilities provided by Chrono to test the previously described MPC controllers on a multibody physics model of a HMMWV. This environment allows a user to simulate their complex multibody HMMWV model on either rigid ground or a customizable granular terrain and test the obstacle avoidance capabilities of their desired controller. This sort of initial algorithm testing would save a controller design team time by allowing for quick execution of high fidelity virtual experimentation to verify initial control algorithm functionality. Then physical experiments on a real vehicle can be performed on controllers that perform successfully in this virtual environment. The overall goal for any controller is eventual successful real-time implementation, so this method of virtual experimentation provides a design team a powerful tool to quicken the process from theoretical controller to physically installed obstacle avoidance controller on a HMMWV. This chapter consists of descriptions of the experiments designed for execution in this virtual environment to analyze the role of model fidelity and influence of terrain type on controlled vehicle performance. The HMMWV, used as the plant vehicle for this study, is described as well as the newly developed method of simulating granular terrain that made this sort of virtual experimentation possible.

#### 3.1 Controlled Chrono HMMWV Vehicle

The model to be controlled (the *plant*) is a full-vehicle Chrono::Vehicle model of a HMMWV, which includes multi-body subsystems for the suspensions, steering,

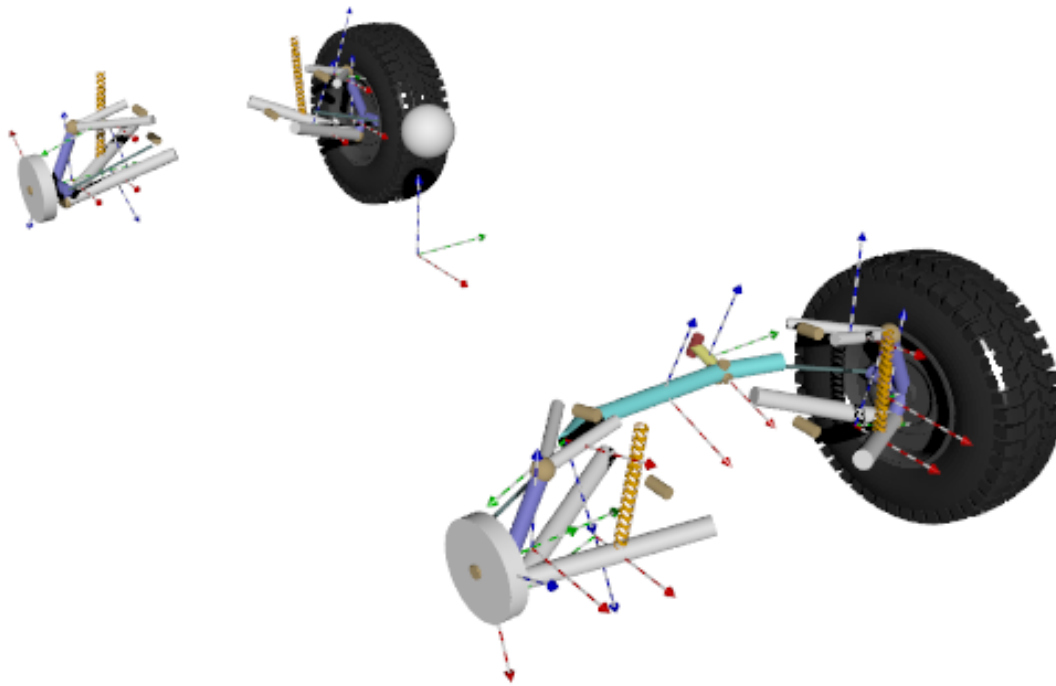


Figure 3.1: Full-vehicle HMMWV multibody model.

driveline, and powertrain, and is available in the **Chrono** package.

This vehicle model (see Figure 3.1) has a curb weight of 2,550 kg. It includes independent front and rear double wishbone suspensions and a Pitman arm steering mechanism. The shock absorber and coil spring, mounted between the lower control arm and the chassis, are modeled with **Chrono** nonlinear force elements and include the effects of bump stops.

The All Wheel Drive (AWD) driveline is modeled using **Chrono** shaft elements and includes three power splitting elements (a central differential and front/rear differentials), as well as conical gears connected through 1-dimensional shaft elements which carry rotational inertia. The powertrain is also modeled using 1-dimensional shaft elements and includes models for a thermal engine specified through speed-torque maps for power and engine losses, a torque converter specified via maps for the capacity factor and the torque ratio, and an automatic transmission

Test Number	Terrain Type	Controller Vehicle Model
1	Rigid	2-DOF
2	Rigid	14-DOF
3	Granular	2-DOF
4	Granular	14-DOF

Table 3.1: Individual Simulation Test Information

gearbox with three forward gears and a single reverse gear. The connection between powertrain and driveline is a force-displacement interface at the driveshaft (with torque applied from the powertrain and angular velocity provided by the driveline).

For mobility studies on deformable terrain, provided that the tire inflation pressure is comparable or larger than the average ground pressure, according to the postulate by Wong [41], the tire can be considered in a so-called *rigid regime*. As such, tire deformation can be ignored and the vehicle’s tires are modeled using rigid contact shapes. The HMMWV tire model used here is thus represented by a cylinder with a radius of 0.47 m (18.5 in) and a width of 0.254 m (10 in).

### 3.2 Simulation Parameters

Simulations on both rigid and granular terrain were compared to understand how the controller performs on non-ideal surfaces. The two internal controller vehicle models studied in these tests were the previously described 2-DOF and 14-DOF vehicle models. All other controller parameters were kept unchanged over all tests. This experiment takes the form of a two level, two factor statistical design of experiments. Referring to Table 3.1, the effect of model fidelity on controller performance was gauged by comparing test 1 with 2 and test 3 with 4, for rigid flat and granular terrain, respectively. Similarly, comparing test 1 with 3 and test 2 and 4 allowed evaluating the performance of each internal vehicle model on rigid and granular terrain.

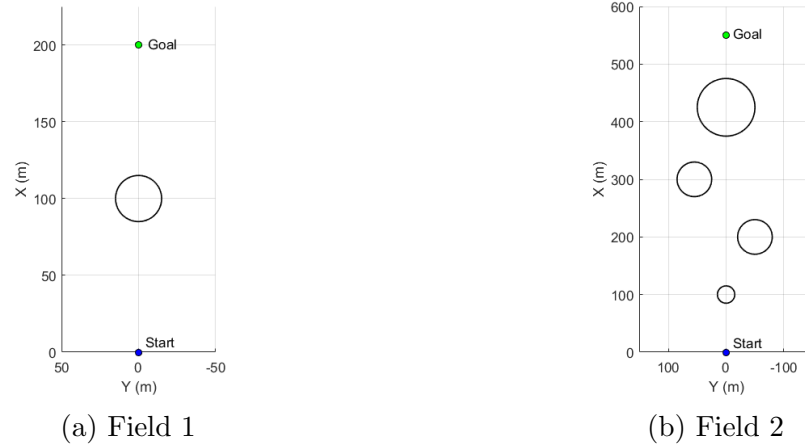


Figure 3.2: Obstacle Fields

	x (m)	y (m)	Radius (m)
Field 1			
Target Location	200.0	0.0	-
Obstacle 1	100.0	0.0	15.0
Field 2			
Target Location	550.0	0.0	-
Obstacle 1	100.0	0.0	15.0
Obstacle 2	200.0	-50.0	30.0
Obstacle 3	300.0	55.0	30.0
Obstacle 4	425.0	0.0	50.0

Table 3.2: Obstacle Field Parameters

Each of the above four tests consisted of runs on two fields with different obstacle distributions. The first one, depicted in Figure 3.2a, contains a single large circular obstacle located along the initial heading of the vehicle, with the target location at a large distance behind the obstacle. The second obstacle field (see Figure 3.2b) consists of four circular obstacles of varying sizes placed in the vehicle's initial heading direction, and provides a more realistic obstacle-avoidance scenario. Table 3.2 summarizes the obstacle locations and dimensions, as well as the target location for the above two scenarios.

The following vehicle parameters are maintained throughout all of the executed tests. A PID speed controller is used to maintain a near constant speed of 8.1 m/s

longitudinally for the simulated plant **Chrono** wheeled vehicle. This constant speed is also enforced in the analytical models internal to the MPC controller. The LIDAR sensor has a maximum range  $R_{LIDAR} = 129.6$  m and is sampled instantaneously at increments of  $\epsilon = 2.5^\circ$ . The vehicle is limited to a maximum steering angle of  $\delta_f = 10^\circ$ , with a maximum steering rate of  $\varsigma_f = 70^\circ/\text{s}$ .

### 3.3 Evaluation Metrics

Five evaluation metrics will be used to compare one test performance to the other. Using the DEM-P method of simulation results in noisy acceleration data. To address this, acceleration data is filtered to remove noise. First, all test runs will be compared based on the time to reach the target,  $T_{target}$ , to determine which controller leads the vehicle to the target point quickest. Second, the closest distance the vehicle reaches to any obstacle,  $d_{min}$ , will be measured. Third, the control effort will be calculated and compared between test cases. The better test case will have a lower controller effort value. Finally, the maximum and average lateral accelerations will be calculated and compared. The lateral accelerations are calculated at the driver's position in the chassis. These five evaluation metrics provide a consistent methodology for comparing one test case with another, regardless of the terrain type and underlying analytical controller model.

### 3.4 Granular Terrain Simulation

Due to the large-scale nature of the simulations in this study, generating granular particles distributed across the entire obstacle field is both computationally exhausting and unreasonable. Instead, a moving-patch approach was developed. Consider an AGV on a large terrain patch of 100 by 100 meters. Since the vehicle behavior and performance is of primary interest and not explicitly the terrain behavior, only a small patch of granular terrain underneath and around the vehicle is assumed significant. This idea promoted the development of a new relocating

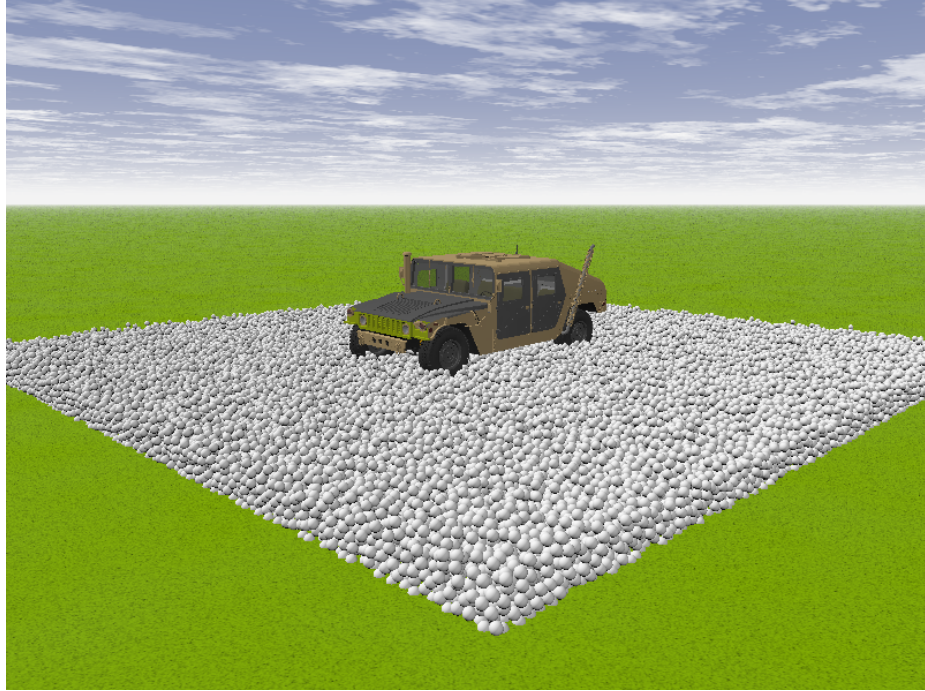
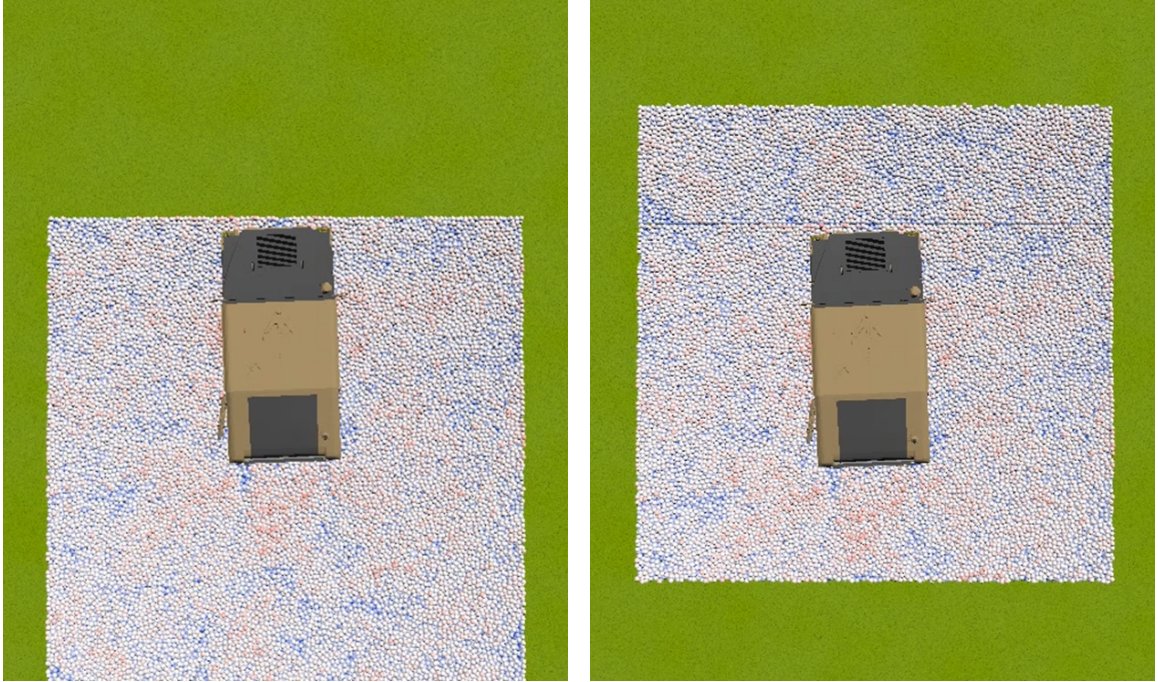


Figure 3.3: Relocating Granular Patch that follows the vehicle

granular patch which enables simulations of a vehicle traversing granular terrain over a large area, without the need to generate particles everywhere throughout that area.

This mechanism allows a user to generate granular particles within a specified distance of the vehicle's CoG. Figure 3.3 presents the patch of granular terrain underneath the **Chrono** HMMWV that moves with the vehicle. The particles are contained within four rigid walls to prevent them from escaping the desired terrain area. As the vehicle moves across the terrain, this patch maintains granular material underneath the vehicle by consistently relocating particles that are too far away from it. When the vehicle location comes within a certain predefined distance of any of the walls, a band of particles half the length of the HMMWV thick from the opposite end of the granular patch are relocated past the closest wall and all of the walls are shifted in that direction. Each of these relocation steps keeps the vehicle on top of granular terrain at all times. Depending on the vehicle, this granular patch can be resized for both computational efficiency and to guarantee the vehicle



(a) Vehicle Approaches front wall just before relocation algorithm triggered

(b) Particles behind the vehicle are relocated to in front

Figure 3.4: Granular relocation algorithm

maintains a reasonable distance from the walls to prevent any boundary effects. This newly developed method of simulating granular terrain for AGV's is omnidirectional in the ground plane, allowing the vehicle to move in any direction while the terrain patch relocates and responds to that movement, thus enabling simulations over arbitrarily large areas. This algorithm was newly developed for the purpose of this research and without which this study would not have been possible. This algorithm is visualized in the x-direction in Figure 3.4.

For these simulations, the granular patch was maintained at a size of roughly 6.6 by 6.6 meters. This number was assigned as two times the largest dimension of the vehicle. The dimensions of the patch are not constant, however, since the patch expands in the direction of relocation by two times the largest particle radius every time the advancing wall is shifted. This extra radius is added on the relocation distance in order to avoid particle overlap, since the relocated particles are moved to a position that is shifted one largest particle radius ahead of the particles adjacent

to the advancing wall. After relocation, the walls of the granular patch are given a recovery velocity such that the granular patch regains its original dimensions in 0.1 seconds. This recovery time should be small relative to the duration of the simulations, and in general it will also depend on the velocity of the vehicle, since the granular patch should completely recover its dimensions between subsequent relocations.

To prevent the granular patch from colliding with the obstacles in the field, each body (particles, ground, containing walls, obstacles) in the simulation is designated a family. Then within the simulation the user can designate which families collide with which other families. So the particles and their walls are all members of the same family. This family interacts only with the family containing the bodies of the HMMWV and not with the obstacle family. This framework allows the vehicle to approach and collide with the obstacles without the terrain actually experiencing that obstacle collision. Figure 3.5 presents the vehicle on the granular patch maneuvering around an obstacle and though the terrain overlaps with the obstacle, no collision forces are calculated or applied between those families of bodies.

Figure 3.4 presents the single granular patch used for all granular simulations in the presented experiments of this thesis. The granular terrain is colored based on the particle global  $z$  location so that blue particles are low and red particles are high. An improved alternative to this single relocating granular patch was explored as well in an attempt to both speed up simulation runtimes and allow for simulation of smaller particles. Instead of simulating a single large granular patch underneath the vehicle center, four individual granular patches are simulated underneath each of the vehicle wheels. This concept is visualized in Figure 3.6. This concept would allow for a much smaller number of particles to be simulated because resources again are not being wasted for particles sitting too far away from the wheels. Each smaller patch works the same as the large single granular patch where they are

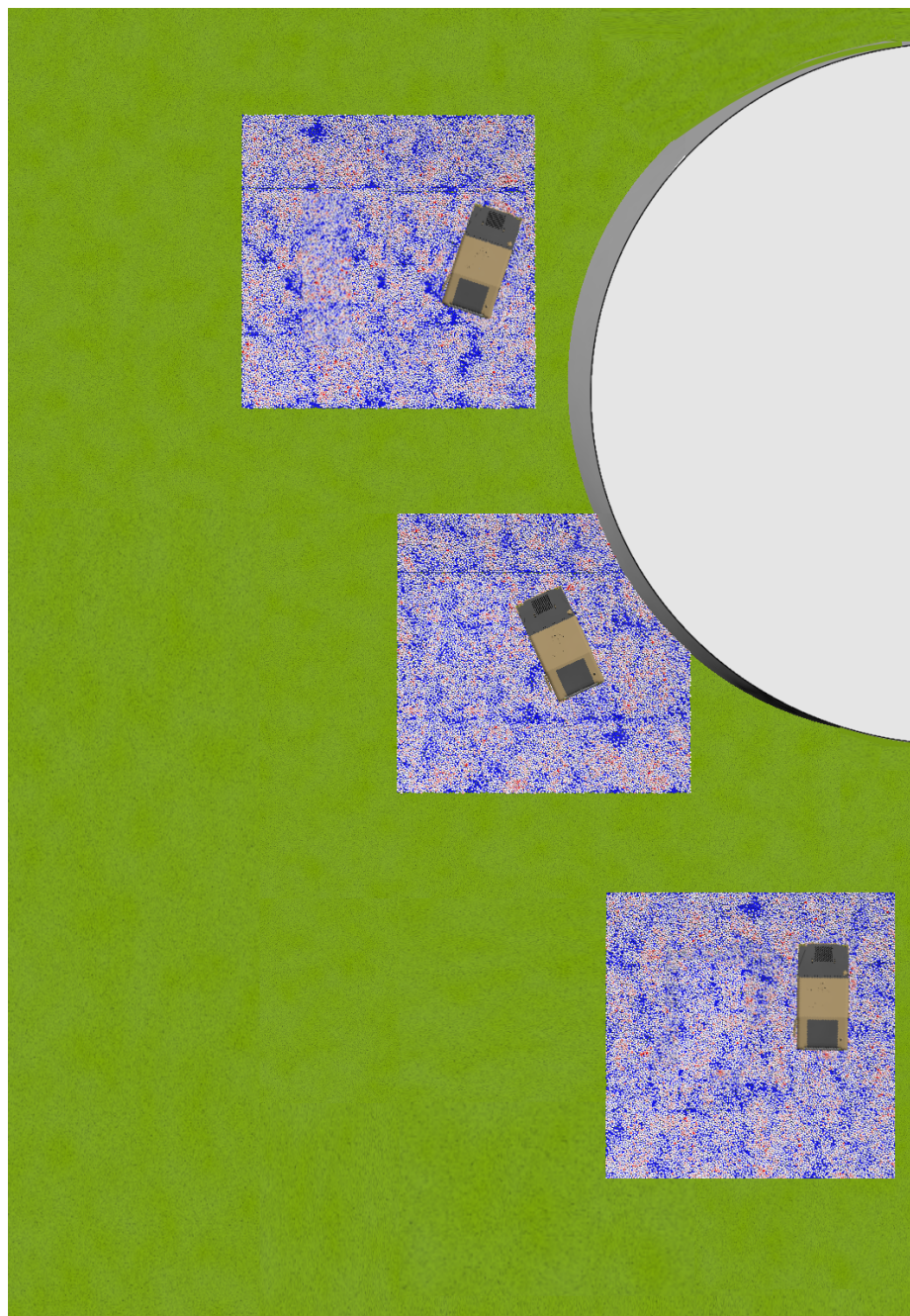
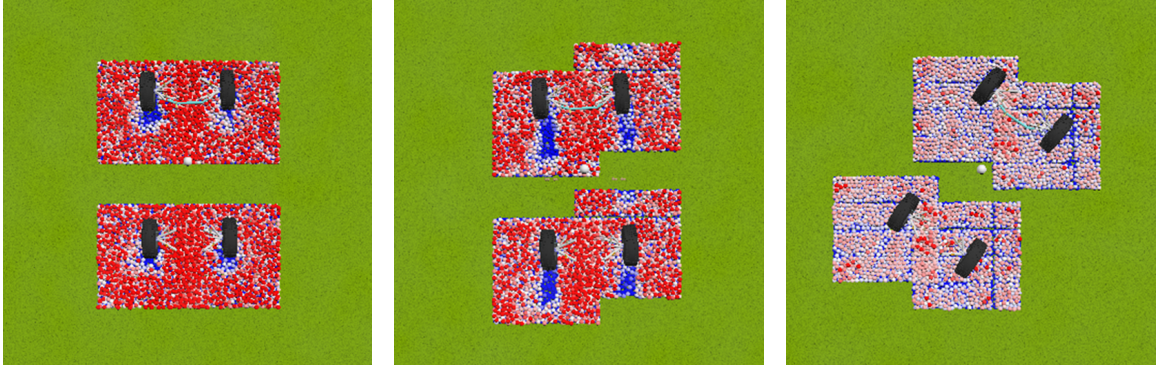


Figure 3.5: HMMWV navigating around obstacle, granular Patch does not collide with obstacles

provided a location on the body to follow. As that point on the body moves a certain distance towards any of the walls holding the small patch of particles together, particles near the opposite wall are relocated to in front of the closer wall, the walls shift and compress to account for the patch growth to prevent particle overlaps during relocation. To prevent particles from one wheel patch from colliding with particles from another wheel patch, bodies are designated as members of a family depending on which patch they were generated in. Particles then only collide with the walls, particles, and wheel that are members of its own family, allowing patches to overlap without actually colliding with each other.

This new concept was implemented in simulations to test its potential as a method for simulating granular terrain moving forward. Comparing the results of simulations using these four individual patches to those that used the single large patch, there were not significant differences in results or vehicle trajectories. The individual patches method even allowed for simulations with particles of diameter 0.05 m without sacrificing larger amounts of simulation time to perform these simulations. One note that supports the use of larger particles was even by reducing the particle size by that large of a factor, there was no significant difference in vehicle performance. However, a concern with using these smaller patches was that there was a much more significant boundary effect each relocation step due to the smaller distance between the tire contact patch and the walls containing the particles. Figure 3.6 presents a visualization of the four independent relocating patches explored. This concept shows promise but for consistency and more confidence in the fidelity of the simulation results, all recorded and presented simulations in this study use the single large relocating patch. The independent patches are considered for future research.

Within the granular patch, the granular terrain was modeled by 55,931 uniform spherical particles, with a micro-scale inter-particle sliding friction coefficient of



(a) Initial four relocating granular patches  
 (b) Forward relocation step for the right wheel patches  
 (c) Patches follow wheels as vehicle heading changes

Figure 3.6: Independent four granular relocating patches explored to improve simulation speeds, HMMWV chassis not visualized

$\mu = 0.8$ , and particle diameter of 0.1 m to model a stiff railroad ballast. Previous studies [42] have shown that a randomly packed assembly of as few as 3,000 - 30,000 uniform spheres will exhibit macro-scale bulk granular material yield behavior (due to inter-particle sliding) that closely matches the Lade-Duncan yield surface, which is a well-established yield criterion in the field of geomechanics, where the macro-scale friction angle  $\phi$  for the bulk granular material can be determined as a function of the inter-particle friction coefficient  $\mu$ .

Referring to Figure 10 of [42], a randomly packed assembly of uniform spheres with an inter-particle friction coefficient of  $\mu = 0.8$  will exhibit macro-scale yield behavior corresponding to a bulk granular material with a macro-scale friction angle between roughly  $35^\circ \leq \phi \leq 40^\circ$  if particle rotation is allowed (6-DOF particles), or with a macro-scale friction angle between roughly  $65^\circ \leq \phi \leq 70^\circ$  if particle rotation is prohibited (3-DOF particles). Since the granular patch used in these simulations contains more than 50,000 uniform spheres, this simulated terrain model is accurately modeling the yield behavior of a true granular material on the macro-scale: either with a macro-scale friction angle of  $35^\circ - 40^\circ$  if particle rotation is allowed, which is typical of a wide range of dry natural and crushed sands [43]; or with a macro-scale friction angle of  $65^\circ - 70^\circ$  if particle rotation is prohibited, which

is typical of crushed or fragmented rock, such as railway track ballast [44].

In the case of the HMMWV, the vehicle tire width is equal to only approximately 2.5 particle diameters in the granular patch which is less than ideal. However, a more suited value of 10 particle diameters per tire width would increase the number of particles in the granular patch to more than 3 million, which would result in prohibitively slow simulations, particularly for events corresponding to physical time durations on the order of 10 seconds or more.

### 3.5 Hardware and Materials

The simulations throughout this research were performed on a Quantum TXR430-512R computer with Scientific Linux 7.2 installed as the primary operating system and an Ubuntu 16.04 virtual machine. The specifications of the computer are outlined in Table 3.3. MATLAB 2015b was used to generate all plots in a post-processing script and determine evaluation metrics from each test case. POVRAY was used to generate visualizations of the performed simulations. Due to the long runtimes of each granular simulation, real-time visualization is not possible or necessary. Therefore, visualization with POVRAY is used. Simulations output large POVRAY data files that describe the body locations, sizes, and orientations at each time step. A set of scripts in Python 2.7 were developed to generate images from the POVRAY output data files and then stitch the images together to create a 60 fps video with a prescribed camera angle. The user could define using the scripts the location and orientation of the camera, what point the camera should look at, and if the camera should follow the vehicle throughout the video. This tool allows for simple and effective development of videos and images as a post-processing method. Overall, all work will be accomplished using this computer and resources accessible on this computer.

Identifier	Basic Information	Additional Information	Quantity
<b>Architecture</b>	Quantum TXR430-512R	<ul style="list-style-type: none"> <li>· Tower / 4U Convertable Chassis</li> <li>· Supports 2x E5-2600 v3 Processors</li> <li>· DDR4 Memory</li> <li>· Supports Up To 3x 5.25IN Bays</li> <li>· 8x 3.5IN Hot-Swap HDD Bays</li> <li>· Up To 4x Double-Width GPU</li> <li>· Up To 1TB DDR4 Memory</li> <li>· Dual Gigabit Ethernet LAN</li> <li>· 2000W Redundant Power Supplies</li> </ul>	1
<b>Processor</b>	Intel Xeon E5-2687W v3	10C, 3.1 GHz 25M, Support DDR4-2133	2
<b>Memory</b>	32GB DDR4	2133MHz LR ECC LRDIMM	8
<b>Primary Storage</b>	512 GB 2.5 SATA III	Internal Solid State Drive (SSD) ** OS DRIVE **	1
<b>Secondary Storage</b>	1.8 TB SAS	12Gb/s 2.5InternalHard-Drive**RAID10**	8
<b>RAID Controller</b>	LSI 9300 MegaRAID SAS 9361-8i (LSI00417) PCI	<ul style="list-style-type: none"> <li>· Express 3.0x8 Low Profile SATA/SAS</li> <li>· 1 GB Cache</li> <li>· High Performance Eight-Port 12 Gb/s RAIDController</li> </ul>	1
<b>RAID Cache</b>	LSI LSICVM02	(LSI00418) CacheVault Accessory Kit for 9361 Series)	
<b>GPU</b>	Nvidia Tesla K80	24 GB GDDR5 384-bit PCI Express 3.0 - Passive Cooling	3
<b>GPU Cooling</b>	Passive Cooling	Passive GPU Kit for 4U 4GP USystem	1
<b>Power</b>	PC Smart UPS X 3000VA	Rack/Tower LCD100-127V	1
<b>Internal Cabling</b>		0.6m Internal Cable SFF8643 to x4 SATA HDD (mini SASHD to SATA data port)	2
		2.5" to 3.5" SATA/SAS6Gb SSD Hard Drive Converter / Adapter / Bracket	9
<b>Additional Components</b>		4U / Tower Conversion Kit / Rail Kit	1
		Windows 8.1 Professional 64Bit	1
		Samsung S24D300H 24IN LCD Monitor	1

Table 3.3: Hardware Components of Computer Used for Parallel Simulations

## 4 FIRST EXPERIMENTAL RESULTS AND DISCUSSION

This section summarizes the results of the four tests listed in Table 3.1. The trajectories for the four performed tests are presented in Figures 4.1a and 4.2a, for obstacle fields 1 and 2, respectively. The associated steering commands generated by the controller are presented in Figures 4.1b and 4.2b. The results of this first experiment are published in [45].

The evaluation metrics for each test, as defined in Section 3.3, are tabulated for each obstacle field in Tables 4.1 and 4.2.

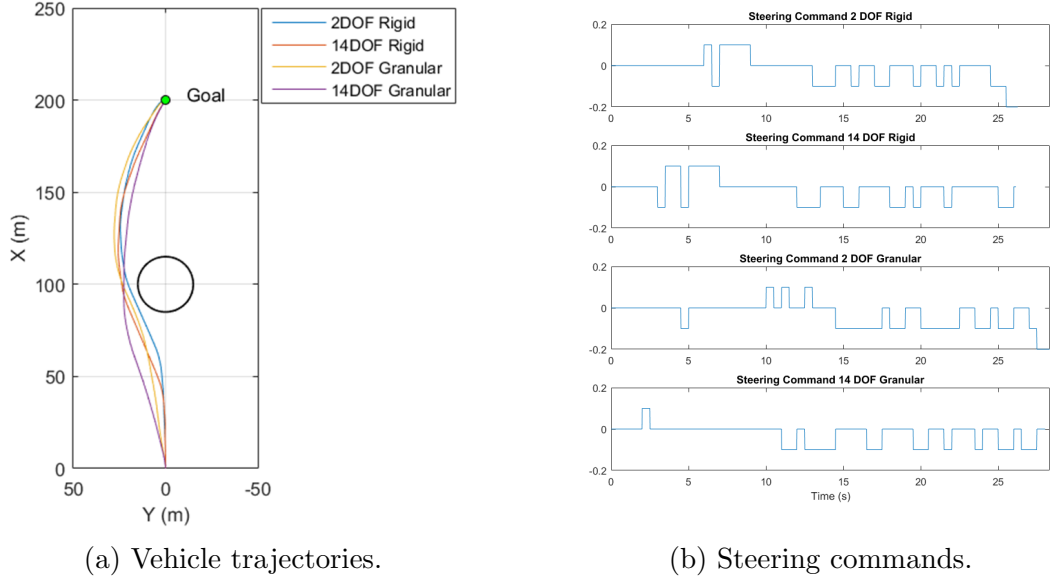


Figure 4.1: Experiment 1 Test results on Obstacle Field 1

Test Number	1	2	3	4
Controller Model	2-DOF	14-DOF	2-DOF	14-DOF
Terrain	Rigid	Rigid	Granular	Granular
Time to Target (s)	26.67	26.15	28.32	28.03
Minimum Obstacle Distance (m)	0.897	5.462	3.491	4.721
Controller Effort	0.0340	0.0340	0.0340	0.0306
Max. Lateral Acceleration ( $\text{m/s}^2$ )	2.78	1.57	2.47	2.33
Avg. Lateral Acceleration ( $\text{m/s}^2$ )	0.54	0.51	0.55	0.46

Table 4.1: Experiment 1 Test Evaluation Metrics Summary on Obstacle Field 1

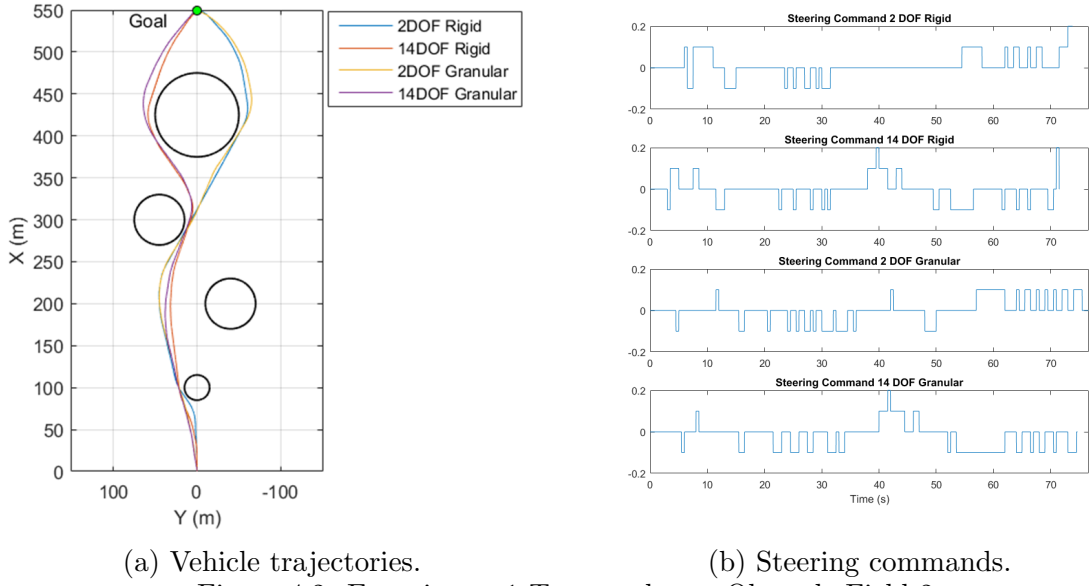


Figure 4.2: Experiment 1 Test results on Obstacle Field 2

Test Number	1	2	3	4
Controller Model	2-DOF	14-DOF	2-DOF	14-DOF
Terrain	Rigid	Rigid	Granular	Granular
Time to Target (s)	73.85	71.55	76.64	74.70
Minimum Obstacle Distance (m)	0.331	2.599	1.083	1.152
Controller Effort	0.0510	0.0680	0.0714	0.0612
Max. Lateral Acceleration ( $\text{m/s}^2$ )	2.92	2.51	2.55	2.45
Avg. Lateral Acceleration ( $\text{m/s}^2$ )	0.41	0.43	0.53	0.58

Table 4.2: Experiment 1 Test Evaluation Metrics Summary on Obstacle Field 2

#### 4.1 Influence of Model Fidelity

Assessing the effect of internal vehicle model fidelity on controller performance when the vehicle navigates on rigid terrain (comparison of tests 1 and 2), the following observations are made:

- the 14-DOF model leads to marginally faster travel to the target location;
- the 2-DOF model leads to trajectories with lower obstacle clearance;
- the two models result in the same controller effort on the first obstacle course; however, the higher-fidelity 14-DOF model requires 33% more controller effort

on the second obstacle course, a consequence of its decision of performing a course change to negotiate the last obstacle;

- the two models result in approximately the same number of command changes on both obstacle courses;
- the 14-DOF model results in lower maximum lateral accelerations.

Overall, while the higher-fidelity internal model leads to marginally better performance, the 2-DOF model is perfectly suitable for use in MPC-based local obstacle avoidance on rigid terrain at non-extreme speeds, as it can safely navigate the vehicle to its end goal. These results support the findings and conclusions in [5] even though a different AGV is considered here.

A similar comparison can be made for the case of obstacle avoidance on granular terrain (tests 3 and 4). As on rigid terrain, the higher fidelity model leads to faster travel to the end goal, larger clearances to the obstacles, and lower maximum lateral accelerations. However, on granular terrain, the controller effort required by using the 14-DOF model is always lower than that required by the 2-DOF model. Therefore, using a higher-fidelity internal controller model leads to overall marginally better performance.

The 2-DOF and 14-DOF internal vehicle models were derived using rigid ground assumptions, yet they are still capable of successfully and safely navigating the simulated vehicle through an obstacle field on granular terrain. The monitored metrics indicate a slight drop in controller performance when using the 2-DOF model. However, these gains are outweighed by the benefits, in terms of required computational effort, offered by implementing the lower-fidelity 2-DOF model.

#### 4.2 Influence of Terrain Type

Turning next to evaluating the 14-DOF internal controller model when navigating on rigid versus granular terrain (comparison of tests 2 and 4), the

following conclusions are drawn:

- on both obstacle courses, the target can be reached faster when navigating on rigid terrain;
- the resulting trajectories are slightly closer to the obstacles when navigating on granular terrain;
- the required controller effort is lower when navigating on granular terrain;
- the maximum lateral accelerations are similar on both types of terrain.

A similar analysis can be performed on the 2-DOF model when used to control a vehicle on either rigid or granular terrain (comparison of tests 1 and 3). Unlike the 14-DOF model, the simpler 2-DOF model does a better job of approaching the obstacles more closely when controlling a vehicle on rigid rather than on granular terrain. Related to this, the controller effort on rigid terrain (test 1) is lower than that required on granular terrain (test 3).

The forces the vehicle experiences from driving on granular terrain can be much different than forces on simple rigid ground. Examining Figures 4.1a and 4.2a, the vehicle navigating on granular terrain does not turn as sharply for a given steering command as the vehicle does on rigid ground terrain. To understand the different behavior between a vehicle driving on rigid and granular terrain, a parametric study should be performed analyzing the vehicle driving on a variety of granular terrains with different granular parameters.

Tests 3 and 4 were performed on granular terrain modeled as randomly packed uniform spheres with an inter-particle friction coefficient of  $\mu = 0.8$ , diameter of 0.1m, and a macro-scale friction angle roughly in the range  $65^\circ \leq \phi \leq 70^\circ$ , with particle rotation prohibited. This granular material resembles railway track ballast. These same tests were attempted on the the same randomly packed uniform spheres

with particle rotation allowed resulting in a macro-scale friction angle between  $35^\circ \leq \phi \leq 40^\circ$ . This granular material resembles a dry sand. The results of this second set of tests on dry sand are not plotted or tabulated because the vehicle failed to move from the initial position. Instead, the vehicle spins its wheels in place at its initial location, making no progress forward as the wheels slowly dig themselves down into the granular terrain. However, this may be due to the too large diameter spheres used for this study for computational reasons. Generally the MPC LIDAR-based constant speed local obstacle avoidance controller used in this study is not yet appropriate for use on all granular terrains. There are situations where a combined speed and steering controller similar to that described in [4], or even some new speed and steering controller which accounts for terrain parameters and mobility information to better predict vehicle movement, would be required. Neither of the models used in this first experiment, the rigid ground 2-DOF yaw plane model nor the rigid ground 14-DOF vehicle models, were appropriate for predicting vehicle behavior and performance on dry sand. Based on these findings, improvements were made to the 2-DOF and controller and are discussed below.

## 5 IMPROVED VEHICLE MODEL ON GRANULAR MATERIAL

Based on the results of the first experiment, the current MPC LIDAR-based constant speed local obstacle controller requires improvements that allow the vehicle and controller to perform better on granular terrain. Using the current internal vehicle models and Magic formula, the controller does not use any information about the terrain to predict vehicle trajectories and estimate tire forces. The current controller would estimate that the vehicle would have the same trajectory on loose soil as on stiff railroad ballast. The controlled vehicle can not in general navigate on multiple different granular terrains because the controller predictions are not influenced by any granular terrain parameters.

This chapter presents the improvements made to the controller by developing a simplified lateral force model for the 2-DOF vehicle model allowing more accurate lateral force estimations at the tire ground contact patch. This new model is derived from terramechanics relations developed by Bekker and Wong. With this new simplified lateral force model, the controller can now take into account the interparticle friction coefficient, friction angle, and cohesion of the soil to better estimate the tractive capabilities of the vehicle. To understand how interparticle friction coefficient affects the trajectory of a vehicle for a given steering sequence, a set of open loop simulations were executed varying interparticle friction coefficient. This set of open loop simulations was also used to verify the new force model does actually predict more accurate vehicle trajectories based on granular parameters. In the future, a MPC algorithm using the 2-DOF model using this new lateral force model will be capable of navigating across multiple different granular terrains by using dynamic terrain data to better estimate vehicle trajectories.

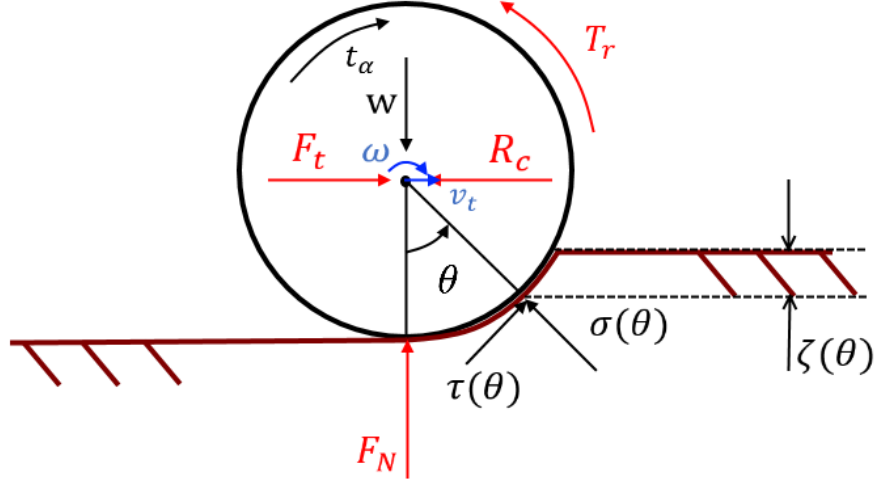


Figure 5.1: Terrain interactions on a single wheel on soft soil

### 5.1 Bekker/Wong Load Sinkage Relationship

To improve the current controller, a new lateral force model was developed using work from Bekker and Wong as motivation. Often the interaction between a rigid tire and soft soil is modeled using terramechanics relations originating from the work of Bekker and Wong [41]. Penetration tests were performed to simulate the contact area of a tire or track interacting with soft ground and the pressure  $p$  and sinkage  $z$  were measured for the array of experiments. Based on this experimental data, Bekker proposed the well known load-sinkage relationship in Equation (5.1) [41].

$$p = \left( \frac{k_c}{b} + k_\phi \right) z^n \quad (5.1)$$

$n$  is an exponent of deformation,  $b$  is the width of the tire,  $k_c$  is the cohesive modulus of deformation, and  $k_\phi$  is the frictional modulus of deformation.

If the normal stress  $\sigma$  is modeled anywhere along the wheel-terrain interface as Bekker proposed, then Equation (5.1) becomes Equation (5.2) where  $\theta$  and wheel

sinkage  $\zeta$  are defined in Figure 5.1.

$$\sigma(\theta) = \left( \frac{k_c}{b} + k_\phi \right) \zeta(\theta)^n \quad (5.2)$$

Using terramechanics relations as in [41], the shear stress  $\tau$  can then be determined along the wheel-terrain interface with Equation (5.3)

$$\tau(\theta) = [c + \sigma(\theta) \tan\phi][1 - e^{-r[\theta_1 - \theta - (1-i)(\sin\theta_1 - \sin\theta)]/K}] \quad (5.3)$$

$$i = \frac{(r\omega - v_t)}{r\omega} \quad (5.4)$$

where  $c$  is the terrain cohesion,  $\phi$  is the internal friction angle of the granular terrain,  $r$  is the wheel radius,  $K$  is the shear deformation modulus, and  $i$  is the wheel slip ratio defined in Equation (5.4).  $\omega$  is the angular speed of the wheel and  $v_t$  is the speed of its center [46].

## 5.2 New Simplified Force Model

The results in Chapter 4 promote the need for a simple yet improved method of estimating the tire forces at the ground tire contact patch when a vehicle is operating on soft or granular terrain. A benefit to the Pacejka Magic Formula presented in Section 2.6 was the tire force calculation was backed by experimental data and the functions were shaped by a large number of experimentally determined parameters. However, the Magic Formula was derived for rigid ground applications only and does not succeed in estimating tire forces on granular terrain [40]. The Magic Formula overestimates the lateral and longitudinal tire forces the vehicle experiences for any given slip angle as it does not take into account the shear strength of the soil. Therefore a simplified method of estimating the lateral tire forces at a tire ground contact patch was developed. This model is then used on the 2-DOF analytical vehicle model to estimate lateral tire forces and better predict

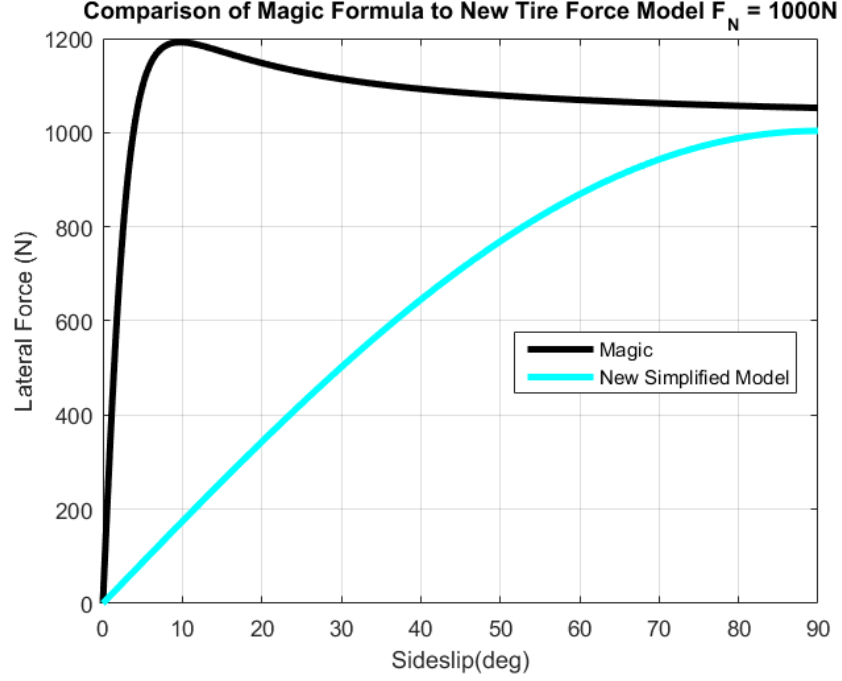


Figure 5.2: Comparison of Magic Formula Lateral Force Prediction to New Force Model

vehicle trajectories on granular terrain.

Rather than using Equation (5.2) to calculate the normal stress at the tire ground contact patch, the normal load is calculated using Equations (2.27) and (2.28). Then the following assumptions are made. First, the tire ground contact patch is assumed to be square of side length  $b$  which represents the width of the tire. Second, a constant normal stress  $\sigma$  throughout the contact patch is assumed such that the total force contribution from this normal stress is equal and opposite the normal load calculated in Equations (2.27) and (2.28) which leads to the simplified estimate of tire ground patch normal stress in Equation (5.5).

$$\sigma = \frac{F_z}{b^2} \quad (5.5)$$

Third, the tangential stress along the tire contact patch is assumed to be constant as well. These assumptions lead to the simplification of (5.3) and when

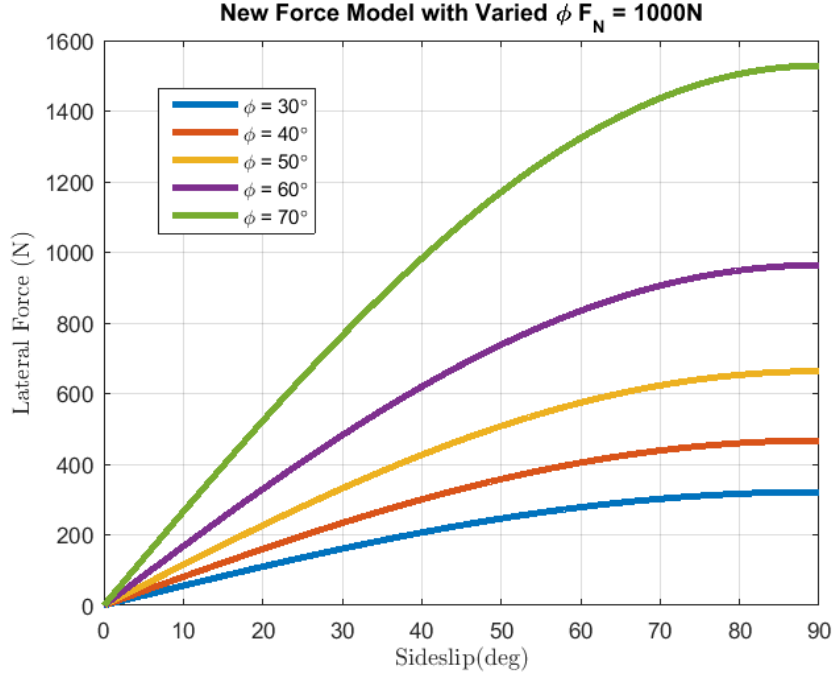


Figure 5.3: Effect of varying internal friction angle  $\phi$  on lateral force prediction

only the pure lateral slip case is considered simplifies to Equation (5.6).

$$\tau = [c + \sigma \tan \phi] [1 - e^{-r[\frac{b}{r} - \sin(\frac{b}{r})]/K}] \quad (5.6)$$

Using the simplified constant stress relations and trigonometry, the lateral force is then calculated using Equation (5.7).

$$F_y = -\tau b^2 \sin(\alpha) \quad (5.7)$$

Though this development is the result of a handful of simplifying assumptions, the results of the formulation should make sense. A comparison between the force prediction as a function of slip as predicted by the Magic Formula and this new force model prediction is shown in Figure 5.2. Intuitively one would expect these results because on granular terrain the tire ground contact patch is subject to a much more limited shear strength than a tire operating on rigid ground. This new

force model then takes into account the cohesion and internal friction angle of the granular terrain in a meaningful way. The new model predicts higher lateral forces when a more cohesive soil or terrain with higher friction angle is used and vice versa as shown in Figure 5.3.

### 5.3 Open Loop Comparisons

To justify that the new force model better models the lateral ground interaction force between a tire and granular terrain, a set of open loop simulations were performed on the HMMWV. In these simulations, the HMMWV was initialized at rest, quickly accelerated to 8.3 m/s, and continued at this speed until the simulation had run for 10 seconds. The vehicle then undergoes the steering sequence presented in Figure 5.4 which consists of the vehicle steering left with a steering angle of 10 degrees for 5 seconds followed by steering right at a steering angle of -10 degrees for another 5 seconds. In this open loop test only the vehicle's trajectory for the last 10 seconds of the simulation is of interest so only this time frame is considered and shown in plots.

Figure 5.5 presents the resulting trajectories of 7 different open loop simulations run with this steering sequence with the parameters just discussed. One simulation was run on rigid flat ground while the other 6 were run on granular terrain consisting of non-rotating particles of inter-particle friction coefficients varying from  $\mu=0.66$  to  $\mu=0.9$ . Lower values of  $\mu$  were considered and tested, but these simulations resulted in the vehicle simply initializing and remaining stuck at the location where it started, constantly spinning its tires without making any forward progress. From these results it is shown that a vehicle operating on granular terrain will steer less sharply than the same vehicle operating on rigid ground undergoing the same steering sequence at the same speed. From these simulations there are clear differences in how a vehicle performs on rigid ground versus on granular terrain and this difference needs to be accounted for when attempting to model

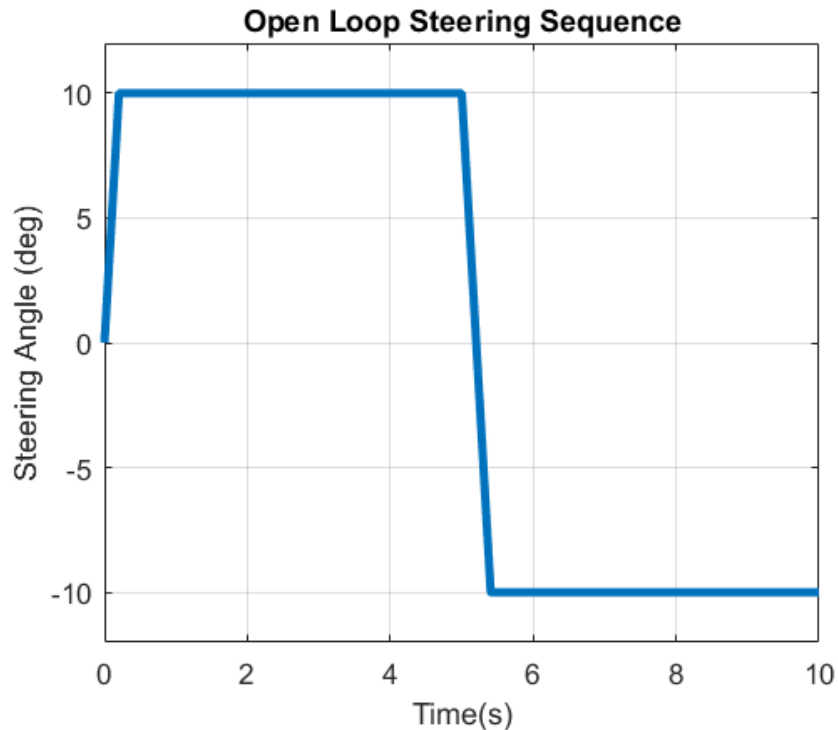


Figure 5.4: Steering Sequence used for Open Loop Simulations

vehicle dynamic performance using simplified analytical equations.

The new force model developed in the previous section seeks to address this difference in vehicle performance based on terrain type. To confirm the new force model does a better job of modeling the lateral force experienced by a tire at the ground tire contact patch, two more open loop simulations were performed. If the 2-DOF model using the new lateral force model does actually do a better job of predicting vehicle trajectories on granular terrain, then the trajectories of the 2-DOF model using the new lateral force model should line up more closely with the trajectories of the HMMWV than the 2-DOF using the Magic formula. This would mean the new lateral force model does a better job of predicting the tractive capabilities of the HMMWV on granular terrain than the Magic formula does.

The 2-DOF analytical yaw plane vehicle model was simulated for 10 seconds while undergoing the same steering sequence shown in Figure 5.4. In one simulation, the 2-DOF model uses the Pacejka Magic Formula to estimate lateral

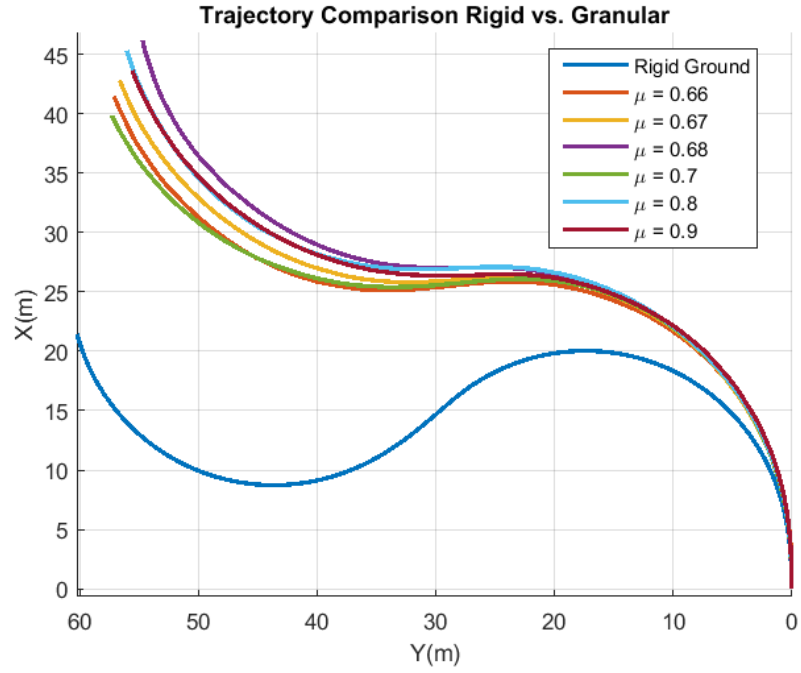


Figure 5.5: Open Loop Trajectory of HMMWV over non-rotating granular terrains with different inter-particle friction coefficients  $\mu$

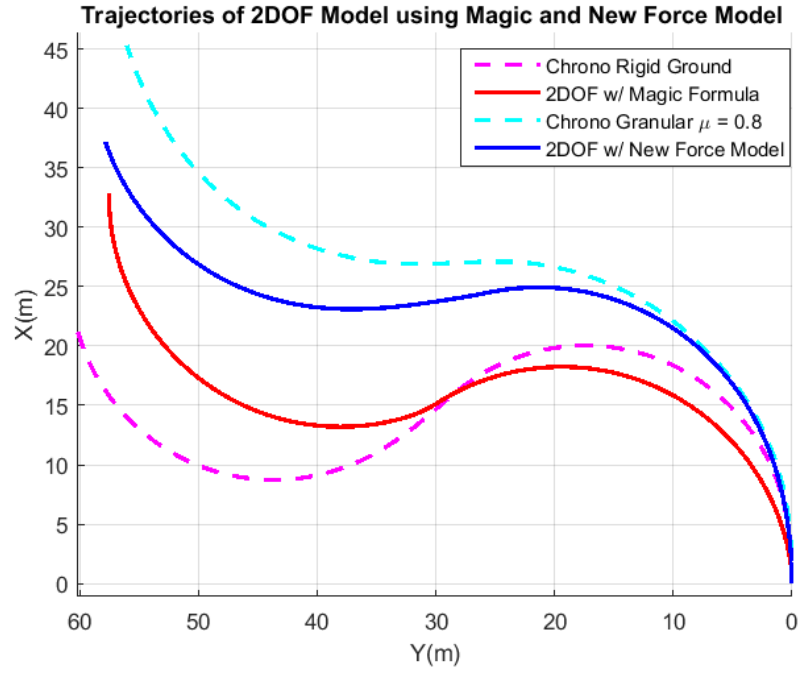


Figure 5.6: Comparison of Trajectory Predicted by 2-DOF model using the Magic Formula to estimate Tire Forces and the 2-DOF model using the new simplified force model

tire forces experienced by the vehicle. The second simulation uses the new force model to estimate the lateral tire forces experienced by the vehicle. These simulation trajectories are presented in Figure 5.6 along with the **Chrono** open loop simulation trajectories on rigid ground and non-rotating granular terrain with  $\mu=0.8$ . The 2-DOF vehicle model using the new force model to predict tire forces matches the **Chrono** simulation on granular terrain much more closely than the 2-DOF model using the Magic formula. Instead the 2-DOF model with the Magic Formula matches the **Chrono** simulation on rigid ground much better. Therefore, the new lateral force model does a better job of predicting vehicle trajectories on granular terrain and even uses granular terrain parameters to better estimate the tractive capabilities of the vehicle. Even though the results in Chapter 4 show the controlled HMMWV was still able to successfully navigate to a target location while avoiding obstacles on granular terrain, there was a clear model mismatch between the internal vehicle controller model and the actual plant vehicle. The 2-DOF model using the Pacejka Magic Formula was overestimating the lateral tire forces because it did not take into account the granular terrain. This new force model does account for the granular terrain parameters and based on Figure 5.6 does a better job of predicting vehicle trajectories on granular terrain.

6 SECOND EXPERIMENTAL RESULTS TO TEST IMPROVED VEHICLE MODEL

6.1 Experimental Setup

Another set of simulations were performed outlined in Table 6.1. This experiment consisted of four simulations on the same two obstacle fields as the previous experiment. Two analytical controller models are considered for this experiment, namely the 2-DOF model using the Pacejka Magic Formula to estimate lateral tire forces on the tires and another 2-DOF yaw plane vehicle model using the newly developed force model to estimate lateral tire forces. As was shown in Section 5.3, open loop simulations enforced that this 2-DOF vehicle model with the new force model does a better job of predicting vehicle trajectories on granular terrain than the previously used Magic Formula 2-DOF models. The same evaluation metrics summarized in Section 3.3 are used in this experiment to compare simulation runs and determine which controller actually worked best. However, the Time to Target evaluation metric in this experiment is deemed the most important metric and therefore this will be the most influential factor in determining which controller worked best because the cost function actively works to minimize the distance to the target each iteration. The other metrics aside from controller effort are not present in the cost function and mainly describe the qualitative conditions of the trajectories.

Test Number	Obstacle Field	Terrain Type	Controller Vehicle and Force Model
1	1	Granular	2-DOF Magic
2	2	Granular	2-DOF Magic
3	1	Granular	2-DOF New Model
4	2	Granular	2-DOF New Model

Table 6.1: Compared Simulations for Second Experiment

6.2 Results

The trajectories for the four performed tests are presented in Figures 6.1a and 6.2a, for obstacle fields 1 and 2, respectively. The associated steering commands generated by the controller are presented in Figures 6.1b and 6.2b. The evaluation metrics for each test, as defined in Section 3.3, are tabulated for each obstacle field in Tables 6.2 and 6.3.

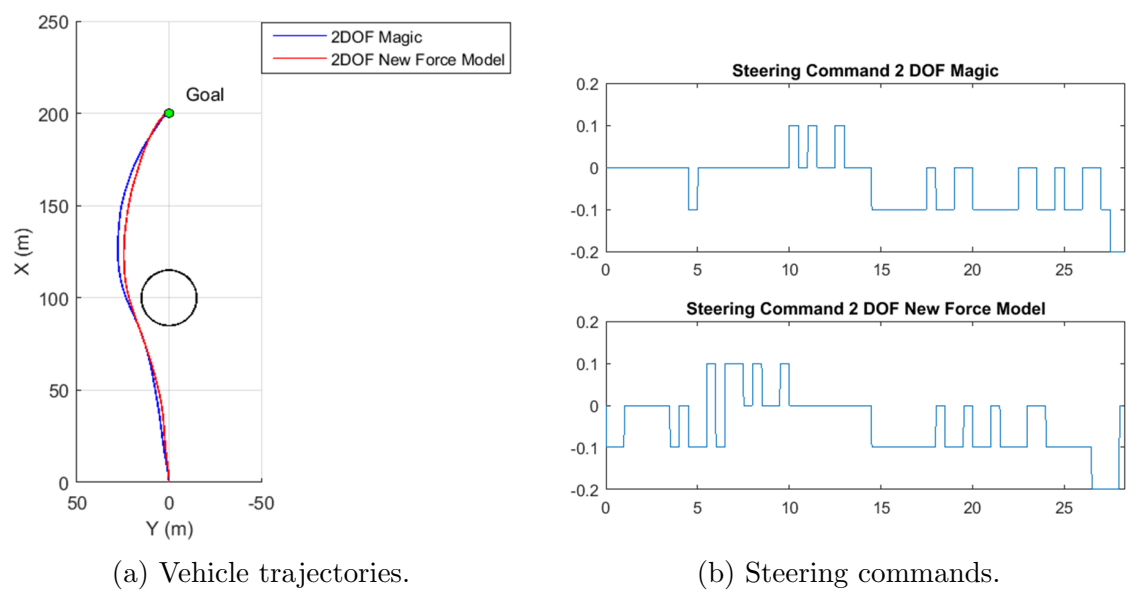


Figure 6.1: Experiment 2 Test results on Obstacle Field 1

Test Number	1	3
Controller Model	2-DOF	2-DOF
Terrain	Granular	Granular
Force Model	Magic Formula	New Force Model
Time to Target (s)	28.32	28.15
Minimum Obstacle Distance (m)	3.491	2.837
Controller Effort	0.0340	0.0381
Max. Lateral Acceleration (m/s <sup>2</sup> )	2.47	2.69
Avg. Lateral Acceleration (m/s <sup>2</sup> )	0.55	0.53

Table 6.2: Experiment 2 Test Evaluation Metrics Summary on Obstacle Field 1

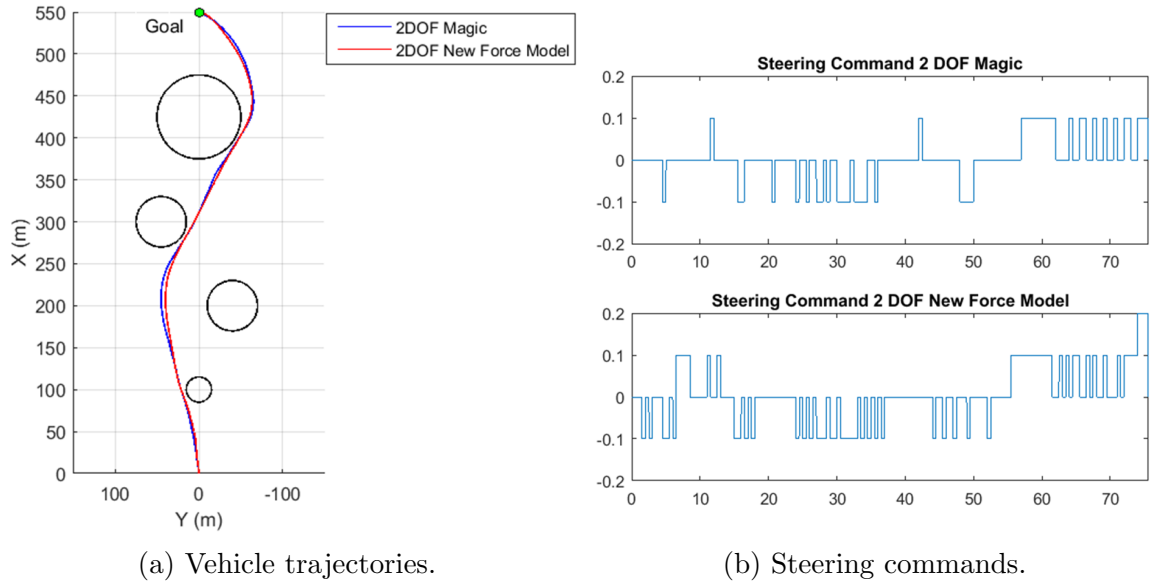


Figure 6.2: Experiment 2 Test results on Obstacle Field 2

Test Number	2	4
Controller Model	2-DOF	2-DOF
Terrain	Granular	Granular
Force Model	Magic Formula	New Force Model
Time to Target (s)	76.64	75.58
Minimum Obstacle Distance (m)	1.083	1.440
Controller Effort	0.0714	0.0952
Max. Lateral Acceleration ( $\text{m/s}^2$ )	2.55	2.75
Avg. Lateral Acceleration ( $\text{m/s}^2$ )	0.53	0.51

Table 6.3: Experiment 2 Test Evaluation Metrics Summary on Obstacle Field 2

From the results summarized in the plots and tables, the 2-DOF model using the new force model to predict lateral tire forces results in the HMMWV navigating to the target quicker than the 2-DOF controller using the Magic Formula to estimate tire forces. This improved performance is witnessed on both obstacle fields. On obstacle field 1, the new force 2-DOF model navigates closer to the single obstacle than the 2-DOF Magic model but the opposite occurs on obstacle field 2. On both obstacle fields, the controller effort of the new force 2-DOF model is slightly higher than the controller effort of the 2-DOF Magic controller. Finally, the average and

lateral accelerations experienced by the vehicle when controlled by each different 2-DOF model are similar and the magnitude differences are negligible.

Qualitatively examining the trajectories of all four tests, the trajectory of the 2-DOF new force model controller appears to be much smoother on both obstacle fields than its 2-DOF Magic controller counterpart which is evident when looking at the runs on the first obstacle field. The 2-DOF Magic model commands result in the HMMWV taking a much wider turn followed by a more aggressive turning maneuver to reach the target position. However, when the vehicle is controlled by the 2-DOF model with the new force model, the HMMWV takes a more direct route to the target position and this improved performance is expressed in the decreased Time to Target. This difference is due to the Magic Model's overestimation of lateral tire forces that the HMMWV experiences. Since the 2-DOF vehicle model using the Magic formula estimates the vehicle can turn sharper at a given steering angle than the terrain actually allows, the predicted trajectories for any given steering angle are wrong and the HMMWV ends up changing its heading less than the controller predicts.

This improvement in the trajectory smoothness can be seen comparing the trajectories on obstacle field 2. Both controllers command the HMMWV to follow similar trajectories, navigating the HMMWV in the same direction around all of the obstacles. Again though the 2-DOF new force model controller trajectory seems smoother of the two. The internal analytical vehicle model matches the plant vehicle better which leads to predicted trajectories that match the actual behavior of the HMMWV. These two facts combined lead to a better controlled HMMWV that navigates the the target quicker than if its trajectories were predicted using the 2-DOF Magic formula model.

Overall based on the comparison of evaluation metrics and qualitative comparison of the trajectories, in this experiment the controller with the 2-DOF

vehicle model using the new force model performed better than the controller using a 2-DOF model with the Magic formula. The trajectories of the controlled HMMWV are smoother and result in quicker Time to Target metrics. It is therefore possible to develop a tire force model which takes into account the soil parameters to estimate the lateral forces at the tire ground contact patch and use this new tire model to better predict vehicle trajectories in a MPC controller. In the future, this 2-DOF vehicle model with the new force model should be used or developed further to better control a wheeled vehicle on flat granular terrain of varying cohesive and shear strengths as well as inter-particle friction coefficients.

## 7 CONCLUSIONS

In this study, using the multibody physics API **Chrono**, a simulation has been developed of a HMMWV driving through a user specified obstacle field towards a defined target location. Within this simulation, a MPC LIDAR based local obstacle avoidance controller has been implemented to navigate the vehicle around obstacles as it encounters them. The controller itself uses a simplified analytical vehicle model to predict the actual controlled **Chrono** vehicle states within a finite prediction horizon in order to determine the optimal path and steering sequence forward from the current vehicle state. A method of simulating granular terrain for this large obstacle field was developed and implemented for this study. Two separate MPC Algorithms were developed initially, one that uses a 2-DOF vehicle model using the Pacejka Magic Formula as a tire model to predict **Chrono** vehicle states and a 14-DOF model to do the same. These two controllers were used to navigate a **Chrono** HMMWV through two obstacle courses on both rigid terrain and then on granular terrain. The results were compared to understand what if any improvements need to be made to the MPC LIDAR based local obstacle avoidance controller to successfully control a vehicle on granular terrain.

As in [5], the controller with the 14-DOF internal vehicle model performs marginally better than the controller with the 2-DOF internal vehicle model in all situations. However, tests with the 2-DOF controller prove that this controller can still successfully navigate a vehicle through an obstacle field when the vehicle is moving at non-extreme speeds. Using the 2-DOF controller also allows for faster calculation of optimal steering sequences which is required for eventual real-time implementation. Both of the controllers navigated the vehicle worse on granular terrain than on rigid ground terrain, but this is expected since the internal vehicle controller models were derived using rigid ground assumptions.

This study also highlights the complexities introduced to vehicle modeling when the terrain is no longer rigid. Looking at the vehicle trajectories and lateral accelerations on granular terrain, there are clear differences in tire ground interactions when the terrain is now granular. The terrain granular parameters affect the turning characteristics of the vehicle, acceleration abilities, and overall vehicle dynamic performance which is not control predictive with the currently used 2-DOF and 14-DOF analytical vehicle models.

From the results of the first experiment, a new lateral tire force model was developed using common terramechanics relationships developed by Wong and Bekker. This new force model uses the cohesive strength and internal friction angle of the granular terrain to calculate the lateral forces experienced by a tire provided the current slip angle and wheel speed. From the first experiment, it was determined that though the 14-DOF internal vehicle model resulted in better controller performance, the complexity of this model made it undesirable for real-time implementation. Though the 2-DOF controller navigated to the target marginally slower, it still did succeed in reaching the target. Therefore the 2-DOF model was deemed desirable for future studies due to its success in reaching the target as well as the quicker runtime speeds necessary to determine an optimal path.

For these reasons then a second experiment was conducted using two different controllers. The first is the same controller from the first experiment consisting of an internal 2-DOF vehicle model using the Pacejka Magic Formula to predict vehicle trajectories. The second controller uses the 2-DOF vehicle model but with the new lateral force model to better predict trajectories on granular terrain. These controllers were tested on the same two obstacle fields from the first experiment but solely on a granular terrain resembling a stiff railroad ballast, the same granular terrain as from the first experiment. The new controller performed better than the old controller using the Magic formula to predict tire forces and this improvement is

shown by a lower Time to Target as well as qualitatively smoother trajectories with the second controller. This new intermediate analytical model is not as complex as the 14-DOF model but does take into account granular parameters such as the internal friction angle to result in more accurate vehicle trajectory predictions on granular terrain and ultimately a better controlled HMMWV.

Each of the five objectives have been accomplished through the development and execution of the experiments described in this study. For clarification, a summary of each of the five objectives and how they were accomplished are presented as follows.

1. Develop a robust simulation environment using **Chrono** to simulate granular terrain efficiently over large terrains without actually modeling every single particle over the large terrain. This was accomplished by leveraging the multibody and parallel computational abilities provided by **Chrono**. A novel relocating soil patch was developed to make simulation of a vehicle traversing large granular terrains possible without requiring months of runtime per experiment.
2. Study the performance of the MPC Controller on granular terrain as compared to that on rigid terrain. The results of this specific performance analysis are summarized in Chapter 4. These results enforced the need for more research into a better method of predicting tire forces at the tire ground patch in a simple manner.
3. Analyze the impact of fidelity of the internal controller model on speed and performance of the obstacle avoidance controller. These results are summarized in Chapter 4 for Experiment 1. Overall similar conclusions were made to those drawn in [5]. Namely, the controller using the more complex 14-DOF model resulted in marginally quicker Time to Target metrics than the 2-DOF controller model, but the increased complexity of the 14-DOF model

increased controller runtimes making the 14-DOF model an undesirable choice for real-time implementation. The 2-DOF controller did still perform successfully and was chosen for more development to improve its performance in Experiment 2. The 14-DOF model should be used to determine safety constraints to impose on a 2-DOF model used as the controller model.

4. Improve the controller for better performance on granular terrain. Chapter 5 presents the development of this improved 2-DOF vehicle model which uses a newly developed lateral force model stemming from Bekker and Wong's terramechanics relations to more accurately predict vehicle trajectories on granular terrain. Chapter 6 presents the results of a second experiment run to confirm the newly developed vehicle model results in better controlled HMMWV performance.
5. Showcase the potential of controller testing in a high fidelity virtual test environment with **Chrono** to assist with initial control algorithm development before physical implementation for vehicular applications. All of the simulations and experiments from this study support and prove this objective. Using **Chrono** to create a virtual testing environment made it possible to quickly change aspects of the controller design to understand specific aspects of the control algorithm such as model fidelity or terrain's effect on controller performance. Testing the controller through simulation allowed for quicker and safer experiments than if one were to attempt to run each of these experiments in real life. Though the goal of these virtual experiments is to eventually prepare the controller for real life implementation and tests, these virtual tests should speed up the design process and make real life experiments that much more beneficial and meaningful to the design team.

Richard Hamming once said the purpose of computing is insight, not numbers

[47]. All mathematical models are an abstraction of reality, so what the user is interested in exploring or modeling should drive the complexity of the model. Detailed analysis of vehicle performance requires sophisticated high fidelity models often leveraging complex contact algorithms and sophisticated tire behavior representations. Control engineers make use of mathematical models to describe the dynamics of the system they are attempting to control. Linearizations, model simplifications, and other assumptions allow the controller to increase its speed while still describing the desired aspect of the system appropriately. This point is important when considering all of the research and work done to achieve the results of this study. Computer simulations can be extremely beneficial to the design process as long as the user has a strong understanding of the background, mathematical models, and methods that have driven the simulations. However, simulation simply cannot and more importantly should not replace all real life experimentation. No matter how complex a developer makes their simulation, it will still always be an abstraction of reality to some degree. However, it can be an extremely powerful tool in the design process, allowing a user to test core functionalities of their mechanical or controller design to gain insight towards how to improve their design. When used in this manner, virtual and real life experimentation can cooperatively contribute to quicker and more effective design processes. If more tests are run virtually then this saves money and time for the team that then runs physical tests on the most promising designs from virtual experimentation.

This applies directly to all of the work and research performed in this study. This research has provided insight into the MPC LIDAR-based local obstacle avoidance controller. Due to this experiments outlined in this thesis, there is now a better understanding of how model fidelity and terrain affect the performance of this researched controller. From the insight gained from the first experiment, the

controller and internal model were improved and tested virtually. The results of the second experiment emphasized that it is possible to develop a simplified vehicle model that predicts its lateral tire forces using simple terramechanics relations and allow the controlled vehicle to better navigate on granular terrain of varying cohesive strength and frictional properties. In these experiments, a high fidelity sophisticated multibody physics model of a HMMWV was used as the plant vehicle due to the goal of this simulation which is to analyze vehicle performance when controlled by different MPC controllers. Inside of the controller then, simplified analytical vehicle models are used which are another level of abstraction in order to quickly predict the trajectory of the simulated HMMWV based on terrain parameters. As simulation methods improve and computational speeds continue increasing, simulation's role in the design process will increase which can lower design costs and improve design speeds as well.

### 7.1 Future Work

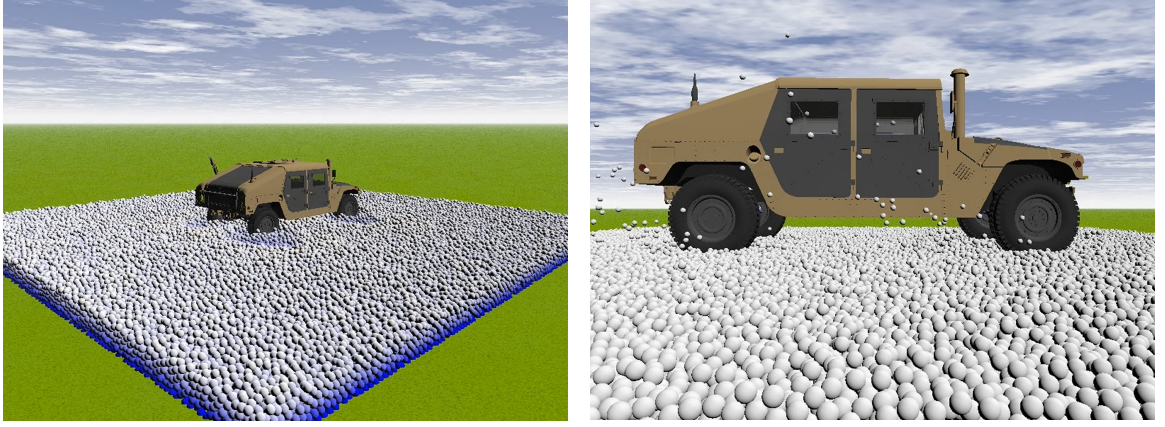
The results of this study are promising. Since the 2-DOF controller is able to navigate the Chrono vehicle on this study's granular terrain, then there is some small set of granular terrain on which a 2-DOF model using the Magic formula is sufficient for predicting vehicle performance well enough to guide the vehicle successfully and safely to a target point. However, a new force model was developed using terramechanics relations to better approximate the lateral tire forces experienced by the 2-DOF vehicle model which results in better approximations of vehicle trajectories on granular terrain.

This research opens up many opportunities and avenues for future research topics. This research lends itself to a variety of different parametric studies to understand how changing aspects of the terrain or controller affects the overall vehicle performance. For example, parametric studies examining controller performance on a wider array of granular terrains would allow for insight into how

granular terrain parameters affect vehicle performance. In this research exhaustive search space was used to find an optimal solution. In the future though, a much quicker optimization method would be necessary to solve the optimal control problem each time step and ultimately implement the controller in real-time applications.

A newer control algorithm could be developed all together with the goal of navigating a vehicle on all sorts of terrains. If instead of using a constant speed internal vehicle model and searching for an optimal steering sequence only, a controller could search for an optimal steering and acceleration input. If this were the case then the terrain parameters would impose constraints on how quickly a vehicle could actually accelerate on the terrain thus constraining the search space and resulting in a much smarter granular terrain navigation controller. To accomplish this, the vehicle would need to obtain information about the terrain either dynamically through sensors or in the form of a data map where the vehicle is able to see what qualities the terrain has at any physical location *a priori*.

Another avenue of research would be to explore improvements to the granular simulations developed in this study. To simulate granular terrain, in this study a novel relocating granular patch mechanism was developed that simulates a significant thickness of granular terrain underneath the vehicle at all times and relocates particles in front of the vehicle as it moves in any direction. This research would not have been feasible without the development of this mechanism, but as with any research topic there are further improvements that could help simulation speeds for these sorts of experiments. One possible improvement would be the implementation of periodic boundary conditions at the walls of the relocating patch to minimize any boundary effects at the tire ground contact patch. The independent four granular patches described in this thesis as well could be explored and tested further as a method to improve simulation speeds. That combined with



(a) HMMWV stuck in terrain of 6-DOF particles

(b) HMMWV remains in initial location spinning wheels and spraying particles backwards

Figure 7.1: HMMWV unable to make forward progress in 6-DOF particle field

periodic boundary conditions could potentially result in a quick and robust method for simulating granular terrain.

Future research can explore the cases where the vehicle remained stuck in its initial location. This occurred when the terrain was modeled as full 6-DOF spheres instead of rotation restricted 3-DOF spheres. When simulations were performed to test vehicle performance on these rotating spheres with a macro-scale friction angle of  $35^\circ - 40^\circ$  which is typical of a wide range of dry natural and crushed sands, the vehicle initialized and dropped onto the terrain normally, but throttle commands simply resulted in the vehicle spinning its tires in place. The vehicle would make no forward progress and instead continue to spin its tires while launching particles backwards. This case is visualized in Figure 7.1.

This behavior provides an interesting research topic. The issue of the vehicle being unable to move from this point can be attributed to four possible different reasons. First, this method of modeling dry sand with 6-DOF particles is incorrect and therefore not accurately simulating dry sand. The potential modeling error is possible since particles are not actually smooth spheres but instead have coarser jagged geometries which would result in more rolling friction. Second, the vehicle tires are not modeled correctly and something other than rigid tires should be used.

Third, the controller does not account for how much tractive capabilities the terrain provides and is simply failing to pull the vehicle out of the sand trap. Finally fourth, all of the modeling in this scenario is correct and this behavior is what one would actually experience if they attempted to drive a HMMWV on loose dry sand aggressively. Research devoted to modeling and simulating dry loose sands as well as developing a controller to navigate the vehicle out and along that terrain would be an important development in the terramechanics and controls community.

The research performed here would benefit greatly from physical tests to compliment the results of this study. Insight would be gained relating to hardware and computational speeds necessary to implement this controller on a physical vehicle. Overall the terramechanics field is a highly active and current area of research. The research performed in this study promotes the opportunity for many equally exciting research opportunities relating to terramechanics, simulation, and controls.

## BIBLIOGRAPHY

- [1] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut. Chrono: An open source multi-physics dynamics engine. In T. Kozubek, editor, *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science*, pages 19–49. Springer, 2016.
- [2] T. Shim and C. Ghike. Understanding the limitations of different vehicle models for roll dynamics studies. *Vehicle System Dynamics*, 45:191–216, 2007.
- [3] J. Liu, P. Jayakumar, J. L. Stein, T. Ersal, and J. L. Overholt. The role of model fidelity in model predictive control based hazard avoidance in unmanned ground vehicles using LIDAR sensors. *Dynamic Systems and Control Conference*, 2013.
- [4] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal. An MPC Algorithm with Combined Speed and Steering Control for Obstacle Avoidance in Autonomous Ground Vehicles. *Dynamic Systems and Control Conference*, 2015.
- [5] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal. A Study on Model Fidelity for Model Predictive Control-Based Obstacle Avoidance in High-Speed Autonomous Ground Vehicles. *Vehicle System Dynamics*, 2016.
- [6] Next-Generation NATO Reference Mobility Model (NRMM) Development. [https://www.cso.nato.int/ACTIVITY\\_META.asp?ACT=6243](https://www.cso.nato.int/ACTIVITY_META.asp?ACT=6243). Accessed: September 21, 2016.
- [7] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
- [8] E. Rimon and D. E. Koditschek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.
- [9] J. Borenstein and Y. Koren. The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288, 1991.
- [10] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *IEEE Intelligent Vehicles Symposium*, pages 417–424, 2009.
- [11] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine*, pages 23–33, 1997.

- [12] O. Brock and O. Khatib. High-Speed Navigation Using the Global Dynamic Window Approach. In *Proceedings of the 1999 IEEE International Conference on Robotics & Automation*, pages 341–346, 1999.
- [13] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn. Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review. pages 537–544, 2006.
- [14] S. Shimoda, Y. Kuroda, and K. Iagnemma. High-Speed Navigation of Unmanned Ground Vehicles on Uneven Terrain using Potential Fields. *Robotica*, 25:409–424, 2007.
- [15] Y. Koren and J. Borenstein. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1398–1404, 1991.
- [16] F. Allgower and R. Findeisen. An Introduction to NonLinear Model Predictive Control. In *21st Benelux Meeting on Systems and Control*, 2002.
- [17] P. Ogren and N. E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics*, 21:188–195, 2005.
- [18] J. Primbs, V. Nevistic, and J. Doyle. Nonlinear Optimal Control: A Control Lyapunov Function and Receding Horizon Perspective. *Asian Journal of Control*, 1(1):14–24, 1999.
- [19] A. Tahirovic and G. Magnani. Passivity-based Model Predictive Control for Mobile Robot Navigation Planning in Rough Terrains. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
- [20] G. P. Bevan, H. Gollee, and J. O’Reilly. Trajectory Generation for Road Vehicle Obstacle Avoidance Using Convex Optimization. *Journal of Automobile Engineering*, 224:455–473, 2010.
- [21] M. Nanao and T. Ohtsuka. Nonlinear Model Predictive Control for Vehicle Collision Avoidance Using C/GMRES Algorithm. *IEEE International Conference on Control Applications*, 2010.
- [22] Y. Gao, F. Borrelli, T. Lin, E. Tseng, and D. Hrovat. Predictive Control of Autonomous Ground Vehicles with Obstacle Avoidance on Slippery Roads. *Dynamic Systems and Control Conference*, 2010.
- [23] A. Gray, Y. Gao, J. K. Hedrick, F. Borrelli, T. Lin, and E. Tseng. Predictive Control for Agile Semi-Autonomous Ground Vehicles using Motion Primitives. *American Control Conference*, pages 4239–4244, 2012.
- [24] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. A Model Predictive Control Approach for Combined Braking and Steering in Autonomous Vehicles. *Mediterranean Conference on Control and Automation*, 2007.

- [25] M. A. Abbas, J. M. Eklund, and R. Milman. Real-Time Analysis for Nonlinear Model Predictive Control of Autonomous Vehicles. *IEEE Canadian Conference on Electrical and Computer Engineering*, 2012.
- [26] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma. Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving. *American Control Conference*, pages 188–193, 2013.
- [27] A. Gray, M. Ali, Y. Gao, J. Hedrick, and F. Borrelli. A Unified Approach to Threat Assessment and Control for Automotive Active Safety. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1490–1499, 2010.
- [28] C. E. Beal and J. C. Gerdes. Model Predictive Control for Vehicle Stabilization at Limits of Handling. *IEEE Transactions on Control Systems Technology*, 21(4):1258–1269, 2013.
- [29] K.S. Ti, B. B. K. Huat, J. Noorzaei, M. D. S. Jaafar, and G. S. Sew. A Review of Basic Soil Constitutive Models for Geotechnical Applications. *Journal of Geotechnical Engineering*, 14:1–18, 2009.
- [30] P. Jayakumar, D. Mechergui, and T. M. Wasfy. Understanding the Effects of Soil Characteristics on Mobility. *Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2017.
- [31] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer, 1988.
- [32] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.
- [33] L. E. Silbert, D. Ertas, G. S. Grest, T. C. Halsey, D. Levine, and S. J. Plimpton. Granular flow down an inclined plane: Bagnold scaling and rheology. *Physical Review E*, 64(5):051302, 2001.
- [34] H. P. Zhang and H. A. Makse. Jamming transition in emulsions and granular materials. *Physical Review E*, 72(1):011301, 2005.
- [35] M. Machado, P. Moreira, P. Flores, and H. M. Lankarani. Compliant contact force models in multibody dynamics: evolution of the Hertz contact theory. *Mechanism and Machine Theory*, 53:99–121, 2012.
- [36] C. O’Sullivan and J. D. Bray. Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme. *Engineering Computations*, 21(2-4):278–303, 2004.

- [37] R. Serban and D. Negrut. Chrono::Vehicle – Template-Based Ground Vehicle Modeling and Simulation. Technical Report TR-2016-10: <http://sbel.wisc.edu/documents/TR-2016-10.pdf>, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison, 2016.
- [38] S.G. Jairo. point in polygon 'by quadrants' very fast algorithm. <https://www.youtube.com/watch?v=f-DIsdvwJ7E>. Accessed: 2016-11-18.
- [39] H. Pacejka. *Tire and Vehicle Dynamics*. Butterworth-Heinemann, 3rd edition, 2012.
- [40] F. Mohammadi. Tire Characteristics Sensitivity Study. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2012.
- [41] J. Y. Wong. *Theory of ground vehicles*. J. Wiley, New York, N.Y., 2nd edition, 1993.
- [42] J. A. Fleischmann, M. E. Plesha, and W. J. Drugan. Determination of Yield Surfaces for Isotropic Non-Cohesive Particulate Materials by the Discrete Element Method. *Geotechnical and Geological Engineering*, 32(4):1081–1100, 2014.
- [43] G.-C. Cho, J. Dodds, and J. C. Santamarina. Particle Shape Effects on Packing Density, Stiffness, and Strength: Natural and Crushed Sands. *Journal of Geotechnical and Geoenvironmental Engineering*, 132(5):591–602, 2006.
- [44] B. Indraratna, D. Ionescu, and H. D. Christie. Shear Behavior of Railway Ballast Based on Large-Scale Triaxial Tests. *Journal of Geotechnical and Geoenvironmental Engineering*, 124(5):439–449, 1998.
- [45] N. Haraus, R. Serban, and J. A. Fleischmann. Performance Analysis of Constant Speed Local Obstacle Avoidance Controller Using an MPC Algorithm on Granular Terrain. *Proceedings of the 2017 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2017.
- [46] Bahareh Ghotbi, Francisco González, József Kövecses, and Jorge Angeles. *Mobility Assessment of Wheeled Robots Operating on Soft Terrain*, pages 331–344. Springer International Publishing, Cham, 2016.
- [47] R. W. Hamming. *Numerical Methods for Scientists and Engineers*. McGraw-Hill, 2nd edition, 1973.

# APPENDIX A: RAY-CIRCLE INTERSECTION ALGORITHM

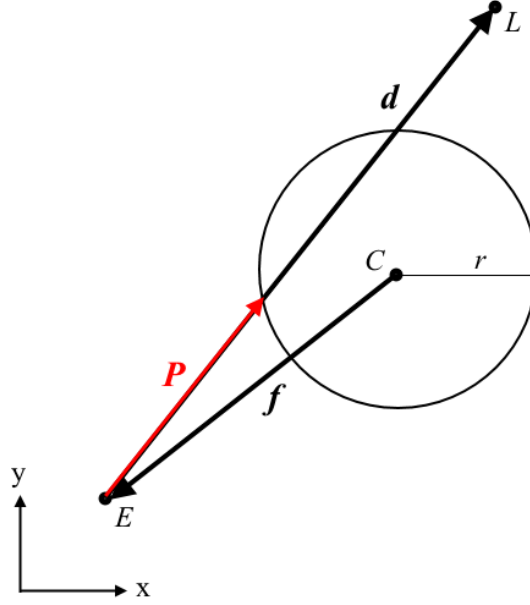


Figure 9.2: Ray-Circle Intersection Visualization

A ray-circle intersection algorithm was implemented in this research to simulate a perfect planar LIDAR sensor and allow the AGV to sense its local surroundings. Consider a two dimensional ray in the x-y plane with the start point  $\mathbf{E}$  and  $\mathbf{L}$  some point along the length of the ray of interest. For this research, this point is the LIDAR radius. Also consider a circle with a center point defined as  $\mathbf{C}$  and radius  $r$ . These points are visualized in Figure 9.2. The following vectors are then formed:

$$\mathbf{d} = \mathbf{L} - \mathbf{E} \quad (9.1)$$

$$\mathbf{f} = \mathbf{E} - \mathbf{C} \quad (9.2)$$

$$\mathbf{P} = \mathbf{E} + t\mathbf{d} \quad (9.3)$$

Here,  $\mathbf{d}$  is a vector describing the ray from its start point to end point.  $\mathbf{f}$  is a vector from the circle center to the start of the vector  $\mathbf{d}$ .  $\mathbf{P}$  is the location of the point where the ray intersects with the circle.  $t$  is a scalar which defines how far

along the length of  $\mathbf{d}$  the intersection between the ray and the circle actually occurs. The circle is then defined by:

$$(x - C_x)^2 + (y - C_y)^2 = r^2 \quad (9.4)$$

$$x^2 - 2C_x x + C_x^2 + y^2 - 2C_y y + C_y^2 - r^2 = 0 \quad (9.5)$$

Equation (9.3) also defines the point or points where the ray intersects with the circle. Noting this yields:

$$x = E_x + td_x \quad (9.6)$$

$$y = E_y + td_y \quad (9.7)$$

Substituting Equations (9.6) and (9.7) into (9.5) results in the following.

$$(E_x + td_x)^2 - 2C_x (E_x + td_x) + C_x^2 + (E_y + td_y)^2 - 2C_y (E_y + td_y) + C_y^2 - r^2 = 0 \quad (9.8)$$

Expanding and grouping like terms yields the following.

$$t^2 (d_x^2 + d_y^2) + 2t (E_x d_x + E_y d_y - C_x d_x - C_y d_y) + E_x^2 + E_y^2 - 2E_x C_x - 2E_y C_y + C_x^2 + C_y^2 - r^2 = 0 \quad (9.9)$$

Equation (9.9) can be simplified further by identifying previously defined vectors and dot products can be substituted in.

$$t^2 (\mathbf{d} \cdot \mathbf{d}) + 2t (\mathbf{f} \cdot \mathbf{d}) + \mathbf{f} \cdot \mathbf{f} - r^2 = 0 \quad (9.10)$$

Equation (9.10) is a quadratic equation that can be solved for two values of  $t$ . The ray-circle intersection algorithm then implemented for this research checks the discriminant of Equation (9.10). If the discriminant is less than zero, then the roots of this quadratic are imaginary and there is no intersection with the circle. If the

discriminant is greater than zero, then there is an intersection and the quadratic formula is used to solve for the two values of  $t$  that satisfy Equation (9.10). Of the two solved  $t$  values, the lower value is checked to see if it is less than one but greater than zero. If so, this  $t$  value is the ray-circle intersection point used as the LIDAR detected point for the algorithm for this specific ray. Otherwise, the algorithm returns the maximum LIDAR radius.

## APPENDIX B: POINT IN POLYGON ALGORITHM

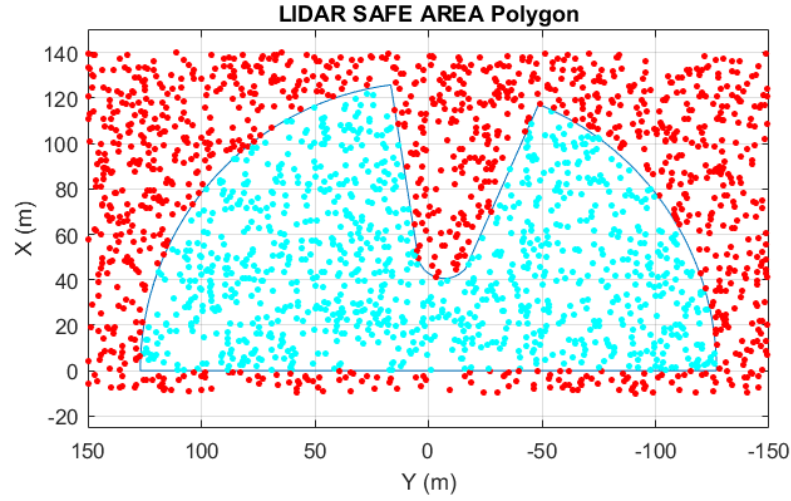


Figure 9.3: Visualization of Point in Polygon Algorithm Working Successfully

The point in polygon was developed to check that all points along a predicted vehicle trajectory fall within the LIDAR safe area represented as a polygon assembled from LIDAR points. To start, the polygon is simply an ordered list of its vertices. Edges are line segments connecting one vertex to the next. The first vertex of the list and the last vertex of the list are also connected to close the polygon. Next, consider a reference frame at the point being tested and express all of the vertices in this relative reference frame. The first vertex is chosen and the quadrant that it resides in is identified. Then the next vertex is chosen and its quadrant is identified as well. If the two points fall in the same quadrant, then nothing happens and the next vertex in the list is checked. However, if the quadrant is different from the previous vertex quadrant, this change is noted and a net quadrant change counter records this change. Clockwise is positive so changes in this direction result in +1 being added to the net quadrant change counter while counterclockwise changes result in -1 being added to the counter. In the case where the quadrant changes by two such as quadrant 1 changes to 3 or 2 to 4 or vice versa, a vector is expressed connecting the old vertex and new vertex. Depending on the direction of

the vector and if the y-intercept is above or below zero determines if +2 or -2 is added to the net quadrant change counter. Once the algorithm cycles through all of the vertices, it checks the original vertex in order to close the polygon. If the net quadrant change is 4 or -4, then the point is in the polygon. Any other situation means the point is outside of the polygon. This algorithm is efficient, simple, and has been implemented successfully for use in this research. However, this algorithm would need to be rethought for application in 3D space. Figure 9.3 presents a set of sample points used to show the algorithm effectively checks to see if the points fall within a sample LIDAR safe area polygon.